Table of Contents

Table of Contents 1
Developer Manual 2
API 3
API Overview 4
API Basics: URL, Methods, Return Formats, Authentication 5
API Errors 6
API Response Examples 8
Generating API Keys 10
Authorizing a Request 11
Retrieving Articles 13
Retrieving Files 14
Retrieving News 15
Retrieving Categories 16
Searching Knowledgebase 17
[v5.5.1] Searching Knowledgebase (v5.5.1 and below) 19

API

The KBPublisher API is designed to allow you to integrate KBPublisher with other applications. These may be desktop or web based, back office or for the public.

API Design

The KBPublisher API is an HTTP based API, meaning that all interactions take place over standard HTTP. For example, GET requests are used to retrieve information and POST requests to submit information.

The API supports 2 different types of return values: XML or JSON. Depending on what client application you're accessing the API from one return format may be easier to use than another. For instance, if you're accessing the API from a javascript library using JSON will be easiest as JSON is javascript. For a desktop client using the API, XML is likely best since that will be the easiest to parse and display. The default return type is JSON.

Access to API resources

Access and allowed actions depend on roles and privileges assigned to user. The same rules applied as in KBPublisher web interface. For example if user who make a API request does not have access to certain articles in web interface then in API these articles never been returned for this user.
For API authentication Public and Private keys are required, each user has his own unique keys.

How to Enable the API

By default API is disabled. You can enable it in Settings -> Admin -> **Enable API access.**
Check **Secure API connection** to configure the API to only accept requests using Secure Sockets Layer (SSL) by using HTTPS. SSL encrypts the transmission, which protects your request and the response from being viewed in transit.

API URL

The API is accessed from the URL where your KBPublisher installed, formatted like so: http://[your_domain]/[kb_dir]/api.php
Example: http://www.domain.com/kb/api.php

API Authentication

The KBPublisher API requires that you authenticate every request by signing it. The process is described in Authorizing a Request.

API Version

KBPublisher version 8.0 supports API versions 1, 2 and 3. By default and if not specified, the API uses recent version.

You can specify the API version you want to use by adding the "**version**" query string parameter to your API URL.
Example: /api.php?call=articles&version=2

Return Format

Two return types are currently supported:

- JSON (default)
- XML

You can specify the return type by adding the "**format**" query string parameter to your API URL. If you want to receive JSON then you do not need to specify the output as that is the default.
Example: /api.php?call=articles&format=xml

Limiting Returned Fields

By default, all available fields are returned when you make a query. You can choose the fields you want returned with the "**fields**" query parameter. This is really useful for making your API calls faster and more efficient.
Example: /api.php?call=news&fields=id,title,datePosted

Example

Here is an example written in PHP
request.php.zip

KBPublisher uses conventional HTTP response codes to indicate success or failure of an API request. In general, codes in the 2xx range indicate success, codes in the 4xx range indicate an error that resulted from the provided information (e.g. a required parameter was missing, etc.), and codes in the 5xx range indicate an error on server, db error or API settings error.

HTTP Status Code Summary

- 200 OK - Everything worked as expected.
- 400 Bad Request - Often missing a required parameter, bad request.
- 401 Unauthorized - No valid API key provided,bad signature.
- 402 Request Failed - Parameters were valid but request failed.
- 404 Not Found - The requested item doesn't exist.
- 500, 502, 503, 504 Server errors - something went wrong on server.

Error Codes

In addition to descriptive error text, error messages contain machine-parseable codes. While the text for an error message may change, the codes will stay the same. The following table describes the codes which may appear when working with the API:

Code	Text	HTTP Code	Description
3	Authentication failed	401	Could not authenticate user.
4	Authorization failed	401	Bad signature, api request could not be authorized.
21	API is available via SSL only	400	Use https to send a request.
22	You cannot access this resource using (%s) request	400	HTTP request methods (POST, PUT or DELETE) is not allowed.
23	Sorry, that page does not exist	400	The specified resource was not found.
24	Sorry, that method does not exist	400	The specified method was not found.
25	Missing or invalid argument(s)	400	Missing a required parameter or invalid parameter.
11	Database error	500	Database error
28	API is not available	503	API is disabled.
29	API is temporarily unavailable	503	API does not respond.
31	Not found	404	The requested item doesn't exist.

If you see an error response which is not listed in the above table, then fall back to the HTTP status code in order to determine the best way to address the error.

Error Messages

When the KBPublisher API returns error messages, it does so in your requested format. For example, an error from a JSON method might look like this:

```
{
   "errors":[
      {
         "errorCode": 25,
         "errorMessage": "Missing or invalid argument(s)",
         "errorInfo": "Required argument(s): timestamp"
      }
   ]
}
```

The corresponding XML response would be:

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
   <error>
      <errorCode>25</errorCode>
      <errorMessage>Missing or invalid argument(s)</errorMessage>
      <errorInfo>Required argument(s): timestamp</errorInfo>
   </error>
```

```
</errors>
```

errorInfo field is an optional in error response.

Get Recent News

Request: /api.php?call=news&method=recent&limit=2&fields=title,link

JSON response:


```
{
   "meta": {
      "page": 1,
      "pages": 1,
      "perPage": 2,
      "total": 2
   },
   "result": [
      {
         "title": "Test News",
         "link": "http:\/\/domain.com\/kb\/index.php?View=news&amp;EntryID=61"
      },
      {
         "title": "Great News Here",
         "link": "http:\/\/domain.com\/kb\/index.php?View=news&amp;EntryID=60"
      }
   ]
}
```


XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<result page="1" pages="1" perPage="2" total="2">
  <entry id="61">
    <title>Test News</title>
    <link>http://domain.com/kb/index.php?View=news&amp;amp;EntryID=61</link>
  </entry>
  <entry id="60">
    <title>Great News Here</title>
    <link>http://domain.com/kb/index.php?View=news&amp;amp;EntryID=60</link>
  </entry>
</result>
```


Get Articles in a Category

Request: /api.php?call=articles&cid=1&fields=id,title

JSON response:


```
{
   "meta": {
      "page": 1,
      "pages": 3,
      "perPage": 2,
      "total": 6
   },
   "result": [
      {
         "id": "131",
         "title": "Quick Response"
      },
      {
         "id": "182",
         "title": "Using Active Directory for Remote Authentication"
      }
   ]
}
```


XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<result page="1" pages="3" perPage="2" total="6">
  <entry id="131">
    <id>131</id>
```

```
    <title>Quick Response</title>
  </entry>
  <entry id="182">
    <id>182</id>
    <title>Using Active Directory for Remote Authentication</title>
  </entry>
</result>
```

Get an Article

Request: /api.php?call=articles&id=1&fields=title,body,tags

JSON response:


```
{
   "result": [
      {
         "title": "API Examples",
         "body": {
            "type": "html",
            "value": "PGgzIGNsYXNzPSJsaW5lVGl0bGUiPjxiciAvPg0KPHNwYW4gc3R5bGU9Im"
         },
         "tags": "api,rest"
      }
   ]
}
```


XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <entry id="10515">
    <title>API Examples</title>
    <body><![CDATA[<h3 class="lineTitle">Get Recent News</span></h3>
<p>JSON response:</p>...
]]></body>
    <tags>api,rest</tags>
  </entry>
</result>
```

Generating API Keys

For API authentication Public and Private keys are required.
Each user has his own unique keys.

To generate and assign keys to user:

- Login to Admin Area
- Click **User** tab
- Click the Options icon under Actions to display the dropdown list, and then click **Details**
- Click **API Settings**, it will open for with API Settings
- Check **API Access** to allow user API requests or uncheck to disalow
- Click **Generate API Keys** if you assign keys in first time or **Reset API Keys** if updating

Important: if you reset API keys it will affect all current scripts for this user

KBPublisher API requires that you authenticate every request by signing it. To sign a request, you calculate a digital signature using a cryptographic hash function. The hash function returns a hash value that you include in the request as your signature.

After receiving your request, API recalculates the signature using the same hash function and input that you used to sign the request. If the resulting signature matches the signature in the request, API processes the request. Otherwise, the request is rejected.

For additional security, you should transmit your requests using Secure Sockets Layer (SSL) by using HTTPS. SSL encrypts the transmission, protecting your request or the response from being viewed in transit.

Arguments to Authorize Request

For any given KBPublisher API request, you must include 3 arguments, it will allow you to authorize the request.
How each value was generated is described below:

accessKey
The accessKey parameter identifies who is making the request.
You can obtain this value from account page in your KBPublisher installation, My Account -> Profile -> API Settings -> Public API Key.

timestamp
The timestamp parameter indicates when the request was created. This value should be the number of seconds since the Unix epoch at the point the request is generated, and should be easily generated in most programming languages. API will reject requests which were created too far in the past, so it is important to keep the clock of the computer generating requests in sync with NTP.

signature
The signature parameter contains a value which is generated by running all of the other request parameters and secret value through a signing algorithm. The purpose of the signature is so that KBPublisher can verify that the request has not been modified in transit, verify the application sending the request, and verify that the application has authorization to interact with API.

Generating a Signature

To produce a signature, start by determining the HTTP method and URL of the request.

- The request method will almost always be GET or POST for KBPublisher API requests.
  HTTP Method - **GET**
- The base URL is the URL to which the request is directed, minus protocol ("http://" or "https://") and any query string.
  URL - **domain.com/kb_directory/api.php**

Next, gather all of the GET parameters included in the request. These values need to be encoded into a single string which will be used later on. The process to build the string is very specific.

- Sort the list of parameters alphabetically by key
- Concatenate params to string like this: key=value&key2=value2... and so on
- URL encode string of parameters

Finally, the signature is calculated by passing the signature base string and signing key to the HMAC-SHA1 hashing algorithm and appended to request. Implementations of HMAC-SHA1 available for every popular language. For example, PHP has the hash_hmac function.

- $signature = rawurlencode(base64_encode(hash_hmac("sha1", $string_to_sign, $secret_api_key, true)));

Here is an example how to sign request in PHP

```
// define api keys
$public_api_key = '1bcf89471d8df298cb6546b1f1da6c8c';
$secret_api_key = '718143f5faw978d6acf5b83c105c27c4';

// collect parameters
$params = array();
$params[call] = 'articles';
$params['accessKey'] = $public_api_key;  // public API key here
$params['timestamp'] = time();
$params['version'] = 1; // optional
$params['format'] = 'json'; // optional

// params to string
ksort($params);
$string_params = http_build_query($params, false, '&');

// collect to string
```

```
$string_to_sign = "GET\n";
$string_to_sign .= "domain.com/kbp_dir/api.php\n"; // api url without protocol
$string_to_sign .= "/\n";
$string_to_sign .= $string_params;

// create signature
$signature = rawurlencode(base64_encode(hash_hmac("sha1", $string_to_sign, $secret_api_key, true)));

// add signature to request string
$string_params .= '&signature=' . $signature;

// ctreate request
$request = 'https://domain.com/kb_dir/api.php?' . $string_params;
```

Request for the above example will be:

http://domain.com/kbp_dir/api.php?accessKey=1bcf89471d8df298cb6546b1f1da6c8c&
call=articles&format=json&timestamp=1385669114&
version=1&signature=LYfL2odFOS4hyJkI5uAZJGD%2BdEM%3D

All API URLs listed here must be prefixed by the root API URL.
Example: http://domain.com/kb/ or https://kb.domain.com/

Retrieve an Article

GET api.php?call=articles&id=[id]

Arguments:

- **skip_hit** (optional) - Do not count entry hit. If this argument is omitted, it defaults to 0.
- **img_rpath** (optional) - Use relative path for images in article body instead of absolute path. If this argument is omitted, it defaults to 0.

List of Articles

GET api.php?call=articles

Arguments:

- **cid** (optional) - The id of the category. If specified, only articles posted to the category will be returned.
- **method** (optional) - Specifies method. If specified, **sort** and **page** argument will be ignored.
  Valid values include:
  - recent (recently updated articles will be returned).
  - popular (most viewed articles will be returned).
  - featured (featured articles will be returned).
- **limit** (optional) - Number of articles to return per page. If this argument is omitted, it defaults to 10. The maximum allowed value is 100.
- **page** (optional) - The page number of results to return. If this argument is omitted, it defaults to 1.
- **sort** (optional) - Specifies how to sort results. If this argument is omitted, it defaults to *date-updated-desc*.
  Valid values include: title-asc, title-desc, order-asc, order-desc, date-posted-asc, date-posted-desc, date-updated-asc, date-updated-desc, hits-asc, hits-desc, rating-asc, rating-desc.

All API URLs listed here must be prefixed by the root API URL
Example: http://domain.com/kb/ or https://kb.domain.com/

Retrieve a File

GET api.php?call=files&id=[id]

Arguments:

- **skip_hit** (optional) - Do not count entry hit. If this argument is omitted, it defaults to 0.

List of Files

GET api.php?call=files

Arguments:

- **cid** (optional) - The id of the category. If specified, only articles posted to the category will be returned.
- **method** (optional) - Specifies method. If specified, **sort** and **page** arguments will be ignored.
  Valid values include:
  - recent (recently updated articles will be returned).
  - popular (most viewed articles will be returned).
- **limit** (optional) - Number of articles to return per page. If this argument is omitted, it defaults to 10. The maximum allowed value is 100.
- **page** (optional) - The page number of results to return. If this argument is omitted, it defaults to 1.
- **sort** (optional) - Specifies how to sort results. If this argument is omitted, it defaults to *date-updated-desc*.
  Valid values include: title-asc, title-desc, order-asc, order-desc, date-posted-asc, date-posted-desc, date-updated-asc, date-updated-desc, hits-asc, hits-desc, rating-asc, rating-desc.

All API URLs listed here must be prefixed by the root API URL.
Example: http://domain.com/kb/ or https://kb.domain.com/

Retrieve News Entry

GET api.php?call=news&id=[id]

Arguments:

- **skip_hit** (optional) - Do not count entry hit. If this argument is omitted, it defaults to 0.
- **img_rpath** (optional) - Use relative path for images in article body instead of absolute path. If this argument is omitted, it defaults to 0.

News List

GET api.php?call=news

Arguments:

- **year** (optional) - Year when news posted. News posted in year equal to this value will be returned. The year should be in format YYYY.
- **method** (optional) - Specifies method. If specified, **sort** and **page** argument will be ignored.
  Valid values include:
  - recent (recent news will be returned).
- **limit** (optional) - Number of news entries to return per page. If this argument is omitted, it defaults to 10. The maximum allowed value is 100.
- **page** (optional) - The page number of results to return. If this argument is omitted, it defaults to 1.

All API URLs listed here must be prefixed by the root API URL.
Example: http://domain.com/kb/ or https://kb.domain.com/

Retrieve Article Category

GET api.php?call=articleCategories&id=[id]

Arguments: none

Article Categories List

GET api.php?call=articleCategories

Arguments:

- **cid** (optional) - The parent category id. If specified, only one level child for this category will be returned.

Retrieve File Category

GET api.php?call=fileCategories&id=[id]

Arguments: none

Files Categories List

GET api.php?call=fileCategories

Arguments:

- **cid** (optional) - The parent category id. If specified, only one level child for this category will be returned.

All API URLs listed here must be prefixed by the root API URL.
Example: http://domain.com/kb/ or https://kb.domain.com/

Search all content

GET api.php?call=search

Arguments:

- **q** (optional) - A free text search. A UTF-8, URL-encoded search query of 1,000 characters maximum, including operators.
- **in** (optional) - Specifies what content to search. If this argument is omitted, it defaults to *all content*.
  Valid values include:
  - all (all content)
  - *article* (search articles).
  - *file* (search files).
  - *news* (search news).
- **by** (optional) - Specifies where to find results. If this argument is omitted, it defaults to *everywhere*.
  Valid values include:
  - all (everywhere - title, article content, keywords/tags)
  - *title* (search in title only).
  - *keyword* (search in keywords/tags only).
  - *id* (search by entry ID).
- **min_date** (optional) - Minimum updated date. Articles with an updated date greater than or equal to this value will be returned.The date can be in the form of a unix timestamp or mysql datetime.
- **max_date** (optional) - Maximum updated date. Articles with an updated date less than or equal to this value will be returned.The date can be in the form of a unix timestamp or mysql datetime.
- **posted** (optional) - Date mode. If specified date posted will be used instead of date updated for above 2 parameters, min_date and max_date.
- **limit** (optional) - Number of articles to return per page. If this argument is omitted, it defaults to 10. The maximum allowed value is 100.
- **page** (optional) - The page number of results to return. If this argument is omitted, it defaults to 1.
- **skip_log** (optional) - Do not log this query. If this argument is omitted, it defaults to 0.

Search Articles only

You can narrow your search by specifying **in** argument.

GET api.php?call=search&in=article

Arguments:

- All arguments listed in **Search all content** section +
- **cid** (optional) - The id of a category to search. If specified, only matching articles posted to the category will be returned.
- **child** (optional) - Specifies whether to search in child categories or not. If specified, articles posted to the category (**cid**) and in all child will be returned, it defaults to 1.
- **custom** (optional) - Specifies whether to search in custom fields. If specified, only articles matching the specified custom values will be returned. You can find custom field IDs in KBPublisher's custom field listing.
  Syntax:
  - custom[custom_field_id]=custom_value
  - custom[custom_field_id]=custom_value_id

  Example:
  - custom[5]=text
  - custom[1]=2
  - custom[4]=2,2

Search Files only

You can narrow your search by specifying **in** argument.

GET api.php?call=search&in=file

Arguments:

- All arguments listed in **Search all content** section +
- **cid** (optional) - The id of a category to search. If specified, only matching files posted to the category will be returned.
- **child** (optional) - Specifies whether to search in child categories or not. If specified, files posted to the category (**cid**) and in all child will be returned, it defaults to 1.
- **custom** (optional) - Specifies whether to search in custom fields. If specified, only articles matching the specified custom

values will be returned. You can find custom field IDs in KBPublisher's custom field listing.
Syntax:
- custom[custom_field_id]=custom_value
- custom[custom_field_id]=custom_value_id

Example:
- custom[5]=text
- custom[1]=2
- custom[4]=2,2

Search News only

You can narrow your search by specifying **in** argument.

GET api.php?call=search&in=news

Arguments:

- All arguments listed in **Search all content** section +
- **custom** (optional) - Specifies whether to search in custom fields. If specified, only articles matching the specified custom values will be returned. You can find custom field IDs in KBPublisher's custom field listing.
  Syntax:
  - custom[custom_field_id]=custom_value
  - custom[custom_field_id]=custom_value_id

  Example:
  - custom[5]=text
  - custom[1]=2
  - custom[4]=2,2

[v5.5.1] Searching Knowledgebase (v5.5.1 and below)

All API URLs listed here must be prefixed by the root API URL.
Example: http://domain.com/kb/ or https://kb.domain.com/

Search Articles

GET api.php?call=articles&method=search

Arguments:

- **q** (optional) - A free text search. A UTF-8, URL-encoded search query of 1,000 characters maximum, including operators.
- **in** (optional) - Search mode. Specifies how to search. If this argument is omitted, it defaults to **article**.
  Valid values include:
  - *article* (search in article title, article body, meta keywords, meta description).
  - *article_title* (search in article title only).
  - *article_keyword* (by tags). Use a comma-delimited list of tags for **q** argument.
  - *article_id* (by article id). Use a comma-delimited list of article id for **q** argument.
  - *article_author_id* (by author id). Use a comma-delimited list of article id for **q** argument.
- **cid** (optional) - The id of a category to search. If specified, only matching articles posted to the category will be returned.
- **cp** (optional) - Specifies whether to search in child categories or not. If specified, articles posted to the category (**cid**) and in all child will be returned.
- **min_date** (optional) - Minimum updated date. Articles with an updated date greater than or equal to this value will be returned.The date can be in the form of a unix timestamp or mysql datetime.
- **max_date** (optional) - Maximum updated date. Articles with an updated date less than or equal to this value will be returned.The date can be in the form of a unix timestamp or mysql datetime.
- **pv** (optional) - Date mode. If specified date posted will be used instead of date updated for above 2 parameters, min_date and max_date
- **limit** (optional) - Number of articles to return per page. If this argument is omitted, it defaults to 10. The maximum allowed value is 100.
- **page** (optional) - The page number of results to return. If this argument is omitted, it defaults to 1.

Search Files

GET api.php?call=files&method=search

Arguments:

- **q** (optional) - A free text search. A UTF-8, URL-encoded search query of 1,000 characters maximum, including operators.
- **in** (optional) - Search mode. Specifies how to search, If this argument is omitted, it defaults to **article**.
  Valid values include:
  - fileall (search in title, filename, file text, meta keywords(tags), description).
  - *file_title* (search in file title only).
  - filename (search in file name only).
  - *file_keyword* (by tags). Use a comma-delimited list of tags for **q** argument.
  - *file_id* (by file id). Use a comma-delimited list of file id for **q** argument.
  - *file_author_id* (by author id). Use a comma-delimited list of file id for **q** argument.
- **cid** (optional) - The id of a category to search. If specified, only matching articles posted to the category will be returned.
- **cp** (optional) - Specifies whether to search in child categories or not. If specified, articles posted to the category (**cid**) and in all child will be returned.
- **min_date** (optional) - Minimum updated date. Files with an updated date greater than or equal to this value will be returned.The date can be in the form of a unix timestamp or mysql datetime.
- **max_date** (optional) - Maximum updated date. Files with an updated date less than or equal to this value will be returned.The date can be in the form of a unix timestamp or mysql datetime.
- **pv** (optional) - Date mode. If specified date posted will be used instead of date updated for above 2 parameters, min_date and max_date
- **limit** (optional) - Number of articles to return per page. If this argument is omitted, it defaults to 10. The maximum allowed value is 100.
- **page** (optional) - The page number of results to return. If this argument is omitted, it defaults to 1.

Search News

GET api.php?call=news&method=search

Arguments:

- **q** (optional) - A UTF-8, URL-encoded search query of 1,000 characters maximum, including operators. Queries may additionally be limited by complexity.
- **min_date** (optional) - Minimum posted date. News with an posted date greater than or equal to this value will be returned. The date can be in the form of a unix timestamp or mysql datetime.
- **max_date** (optional) - Maximum posted date. News with an posted date less than or equal to this value will be returned. The date can be in the form of a unix timestamp or mysql datetime.

- **limit** (optional) - Number of news entries to return per page. If this argument is omitted, it defaults to 10. The maximum allowed value is 100.
- **page** (optional) - The page number of results to return. If this argument is omitted, it defaults to 1.