The Knowledge Base API

The Knowledge Base API

The Knowledge Base offers a REST API for easy integration with other services. Version 1 of this API exposes page content and labels. Some queries are open, others require authentication. Authentication is accomplished via bearer tokens in HTTP headers. If you are interested in obtaining a token for your application, please contact the KB Team.

Interacting with the API

To familiarize yourself, consider using the "curl" command-line utility on Linux or Mac OS X. It can be invoked like this:

```
curl -X [REQUEST] -H '[HEADER]:[VALUE]' [URL]
```

where

- `[REQUEST]` is the HTTP request type (e.g. `GET`)
- `[HEADER]` and `[VALUE]` are the name of a header (e.g. `Authorization`) and the content of that header (e.g. `bearer 12345678`). If you need to specify multiple headers, you may pass as many `-H` options as you need. (e.g. curl -X GET -H 'Thing:One' -H 'Fish:Red' ...)
- `URL` is as described below.

URI and prefix

The API is currently served from: `https://knowledgebase-api.mit.edu/api/`

All endpoints (see below) in the API are preceeded by `v1/` to indicate this is version 1 of the API.

For example, the API URL `https://knowledgebase-api.mit.edu/api/v1/articles/3907262` consists of the **API location** (`https://knowledgebase-api.mit.edu/api/`), the **version** (`v1/`), and the `articles` **endpoint**, along with an argument to pass to it.

Authentication

To authenticate to the API, present your token (see below to request a token) in an Authorization header. For example, if your token is **`12345678-90ab-cdef-1234-567890abcdef`**, you would construct the header as follows:

```
Authorization: bearer 12345678-90ab-cdef-1234-567890abcdef
```

Experienced users will note that this is not in fact an OAuth bearer token – this is correct. OAuth support is planned in a future release, and the Authorization header was designed that way for future compatibility.

Using curl, you would type `curl -H "Authorization: bearer 12345678-90ab-cdef-1234-567890abcdef"`. Using XMLHttpRequest, you would call `request.setRequestHeader('Authorization', 'bearer ' + token)`. For other clients, consult your documentation.

You may pass a token even for endpoints which do not require authentication, however, any tokens you pass must be valid. That is, if you choose to pass a token for an non-authenticated endpoint, and the token is invalid, you will receive a 401 error.

Some endpoints have optional authentication. That is, you can retrieve some content anonymously, but may need to pass a token to retrieve other content. (You will receive a 401 error for the content in question if that's the case.)

Requesting Tokens

API tokens can be requested at https://knowledgebase-api.mit.edu/keys (MIT authentication required), and will automatically be enabled for all public content. Anyone who requires access to protected content in the KB should contact kb-tech-core@mit.edu to discuss their needs.

Content Type

The API returns JSON data, except where noted. Some endpoints are capable of returning HTML or JSON: To obtain HTML, you must set an `Accept:` header with `text/html` as the only value. If it includes any wildcards, you will be given JSON output. If you pass an `Accept:` header that is only capable of `text/html` to an endpoint only capable of generating JSON, you will receive a 406 error.

Return Codes

In the spirit of REST, this API makes full use of HTTP status codes. You are encouraged to make use of them. For example, to see if a resource exists, you may issue a `HEAD` request. If you receive a 404, the resource does not exist. Obviously, a 200 or a 201 will be returned upon success.

For all return codes, you will get a JSON object back, which contains some metadata about the error that occurred. Many objects may contain an `html` attribute, which includes an HTML version of the error suitable for displaying in a browser.

Known Issue: The HTML in error message contains escaped HTML tags, causing the resulting error to display inline HTML tags. Use the `description` attribute of the JSON object in the interim until this is fixed.

Code	Meaning
201	Created (in response to PUT or POST)
400	Bad request. Some aspect of the request was not acceptable.
401	Authorization needed. You will get this if: a) a token is required and you do not pass one; b) your token has expired; c) you pass an invalid token.
403	Forbidden. You successfully authenticated, but your token does not allow access to the resource (either at all, or because you tried to write to read-only resource).
404	Not found. The resource does not exist.
405	Method not allowed. You tried to access an endpoint via a method it does not support. (e.g. POST instead of GET). The response body will contain a `valid_methods` key, indicating what methods will work.
406	Not acceptable. You specified an `Accept:` header that the API cannot honor. (For example, you request only HTML from an endpoint that can only generate JSON.)
500	Internal error. Something bad happened.
501	Not implemented. You found a feature we haven't advertised yet, but we're considering adding. Good for you!

rest_uri

Most objects returned include an attribute called `rest_uri` which is the REST URI of the object ~~relative to the root of the API~~ **This may change.**. For example, if you queried a label to see which articles it contains, each of the objects you received back would contain a `rest_uri` along the lines of `/articles/12345`

Endpoints

/

The root level of the API.

/shortcode

Method	Path
GET	/shortcode/<code>

Parameters	Meaning
<code>	An alphanumeric shortcode

Return a Shortcode object, with information about a shortcode. A shortcode is the series of characters (including any trailing slashes) after the `http://kb.mit.edu/confluence/x/` in a short URL to a page.

Authenticated: no

/articles

Method	Path	Action
POST	/	Create a new article
GET	/<id>	Retrieve an Article object
GET	/<id>/<format>	Retrieve an article in the specified format
GET	/<id>/<format>/excerpt	Retrieve an excerpt of an article in the specified format
GET	/<id>/labels	Retrieve a list of Label objects for each label the article has
POST	/<id>/labels	Add a label to the article by passing 'name=value' in the postdata
GET	/<id>/labels/<label_name>	Get information about the label '<label_name>', if the article has that label
PUT	/<id>/labels/<label_name>	Add '<label_name>' to the article

Parameters	Meaning
<id>	The numeric id of the article
<format>	One of 'object', 'html', 'div'. See below.
<label_name>	The name of a label

Authentication: optional

Format and Excerpt

The default format is `object`, which returns JSON. The `div` and `html` formats are capable of returning `text/html` content, if requested by the `Accept:` header. When JSON is requested, the JSON will have a single attribute (`html`), which contains the HTML.

The `html` format generates the same HTML you'd see if you viewed the article in a browser, complete with references to the KB server. The `div` format generates the page content inside a `<div>` with the DOM id `ConfluenceContent` with minimal formatting.

`/excerpt` may be appended to a request for a formatted article, indicating you want the article's excerpt (enclosed in the Confluence {excerpt} tags), not its full content. If the article does not have an excerpt, the API will return 404. If `object` format is specified, the page object will have the `excerpt` attribute and will NOT have the `content` attribute.

/labels

Method	Path	Action
GET	/labels/<name>	Retrieve a Label object corresponding to the labe of that name.
GET	/labels/<id>/articles	Retrieve a list of ArticleSummary objects that have the label of that id number
GET	/labels/<name>/articles	Retrieve a list of ArticleSummary objects that have the label of that name

Parameters	Meaning
<id>	The numeric id of a label
<name>	The name of a label

Authentication: optional

Objects

Label

Fields	Content
id	Numeric id of label
name	Name of label
rest_uri	URI of Label object

Shortcode

Fields	Content
id	Numeric id of article referred to by shortcode
rest_uri	URI of Article object

ArticleSummary

Fields	Content
id	The id number of the page
rest_uri	URI of this Article object
url	The URL to access the article in the KB
title	The title of the article

Article

And article contains all the fields of ArticleSummary, in addition to:

space	The name of the space the page is in
contentStatus	'current'
created	The creation timestamp
modified	The last modified timestamp
short_url	The short url to the article in the KB

And finally, an Article will contain **one** of the following, depending on how it was requested.

content	The full body of the page
excerpt	The excerpt of the page

Knowledge Base API Proof of Concept

Knowledge Base API Proof of Concept Drupal Module