How to work with APIs

Last Modified on 04/03/2024 12:32 pm EDT

This article provides a basic introduction to APIs, focused on REST APIs. It assumes no prior knowledge. For an introduction to the KnowledgeOwl API, refer to Using the KnowledgeOwl API, and for KnowledgeOwl API reference documentation, refer to Endpoint reference.

This article is designed to be read through in order. Some later sections rely on knowledge from earlier sections.

What is an API?

API stands for "application programming interface".

In basic terms, APIs allow computer programs to talk to each other. The process is broadly the same regardless of the type of API:

> 1. **A client application initiates an API call** to retrieve information—also known as a *request*. This request is processed from an application to the web server via the API's Uniform Resource Identifier (URI) and includes a request verb, headers, and sometimes, a request body.
> 2. **After receiving a valid request**, the API makes a call to the external program or web server.
> 3. **The server sends a *response*** to the API with the requested information.
> 4. **The API transfers the data** to the initial requesting application.
>
> *- from How an API works by IBM*

APIs communicate between programs in a similar way to how people communicate in the physical world. Consider a scenario where you are the writer maintaining your product's knowledge base, and the following set of interactions occur:

1. A customer comes looking for information and can't find it, so they send a support request.
2. In turn, support contact you and ask you to produce an article with the information.
3. The request from support didn't contain enough information for you to be sure what the requirements are, so you reply to them asking for more details (this is similar to an API sending an error message when it receives a badly-formed request).
4. Support send over the necessary information. You can now prepare an article.

5. You need more detailed information about the product, so you contact a subject matter expert who can give you the details you need.
6. The SME sends over the information you requested.
7. You take the information from the SME (and your own knowledge), and write an article that meets the requirements sent by the support team.
8. You contact support and let them know the article is published.
9. Support respond to the customer, providing the article.

Each of those interactions is like an API call, requesting or sending information.

APIs are everywhere. Consider a common example - making an online purchase. This process could involve several APIs to communicate between different services and companies:

- When you first go to the shop, an API may be used to load the product list and information.
- There may be another API handling authentication - perhaps using an external provider, such as logging in using Google.
- Paying for your purchases almost certainly involves another API, this time one belonging to the payment service. That service in turn likely uses your bank's API.

APIs also simplify development. For example, say you want to write a program that interacts with KnowledgeOwl. Our API provides a defined, standardized way for your program to communicate with KnowledgeOwl. You don't need to know exactly how KnowledgeOwl works behind the scenes. Your program doesn't directly access our servers or databases - instead, it calls our API. This is simpler and more secure.

There are different types of APIs. This article focuses on web APIs, and in particular REST APIs.

What is a REST API?

REST stands for "representational state transfer".

A REST API is a type of API designed for use on the web. It can perform actions such as getting or updating information in a database. It uses HTTP verbs, or methods, to indicate the type of action.

HTTP is a key piece of the internet. It is a protocol that allows web browsers and servers to talk to each other. Read more about HTTP in the Mozilla docs.

The methods used in the KnowledgeOwl API are:

- GET - fetch data. For example, you could GET a list of all the tags in your knowledge base.
- PUT - update data. You could use this to update the status of an article, for instance.
- POST - create a new object or piece of data. For example, create a new category.
- DELETE - remove an object. For example, delete an article.

The KnowledgeOwl API can be treated like REST API, although it does not strictly conform to REST API behavior at all times. Refer to Using the KnowledgeOwl API and our Endpoint reference for information on our API.

Calling an API: terminology and tools

Accessing an API, using GET, POST, and so on, is known as **calling** the API. To call an API, you make a **request**. You need to know the **endpoint** you want to use. An endpoint usually provides functionality around a particular **object**, allowing you to access and update its **parameters**.

Request: an API request is a call to a server, using the format required by the API.

Endpoint: in a REST API, the endpoint is a URL. In the KnowledgeOwl API, we have endpoints such as https://app.knowledgeowl.com/api/head/article.json, (which lets you work with articles), https://app.knowledgeowl.com/api/head/tag.json (which lets you work with tags), and so on.

Objects and **parameters**: an object can be thought of as a collection of parameters. For example, the article object parameters include things like project_id (the ID of the knowledge base that the article belongs to), name (the article title), and category (the ID of the category that the article belongs to). These parameters may correspond to database fields.

There are several ways to call an API:

- From your code: for example, when building a website, you might use JavaScript to fetch data.
- With a GUI tool like Postman: there are lots of tools out there to work with APIs. Postman is used to design and test APIs. You can also use it to quickly make an API call.
- Using a command line tool like curl: if you have curl installed on your computer, you can open a terminal (such as PowerShell or Cmd on Windows, or the Mac Terminal) and enter a curl command to call an API.

This guide will briefly introduce Postman and curl.

Calling an API: curl

Let's look at how to make an API request using curl.

If you want to follow along, you need an API key. An API key is a way of authenticating API users. It makes sure that only people with the key can access the data.

We'll start by going through the steps to make a request, then take a closer look at the elements of the request.

It's a good idea to create a separate knowledge base, or at least a hidden test area in your existing one, to experiment with. The example here only retrieves information, so it shouldn't make any changes, but it's always safer to learn and experiment away from your live knowledge base.

If you are on Windows, there are two issues with using curl:

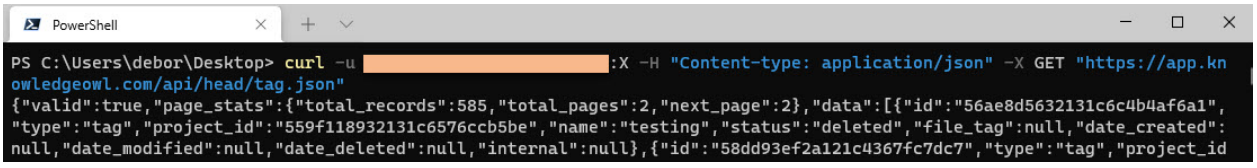
Curl is only available using Command Prompt or PowerShell Core (PowerShell 6 or 7). If you run curl --version and get an error "curl : The remote name could not be resolved: '--version'" you are probably using PowerShell 5 or earlier. Try using PowerShell Core.

By default, curl is aliased to Invoke-WebRequest, a PowerShell cmdlet (command). This can cause errors (for example, if you just run the example in the instructions below, you will receive a 500 error due to malformed data in the section after -d)

1. First, check you have curl installed: open your terminal and type `curl --version`. Hit `Enter`. If curl is installed and working, it should tell you some information about the version you have. If curl is not installed, you can download it.
2. We're going to get a list of all the active (available) tags in your knowledge base. If you don't have any tags, create a couple to use as a test.
3. Copy this curl request into your terminal. It will get all the active tags in your knowledge base:

```
curl -u <API key>:X -H "Content-type: application/json" -X GET "https://app.knowledgeowl.com/api/head/tag.json" -d '
```

4. Replace `<API key>` with your API key.
5. Hit `Enter.` You should see a list of all the tags in your knowledge base. It may take a moment to appear, especially if you have a large number of tags. It should look something like this:



Let's take a close look at the details of the API request:

```
curl 1
-u <API key>:X 2
-H "Content-type: application/json" 3 4
-X GET "https://app.knowledgeowl.com/api/head/tag.json" 5 6 7
-d '{"status":"active"}' 8 9
```

Sample call

1. `curl` tells the command line to use the program called curl.
2. `-u` tells curl that the next bit of information relates to the user. It is followed by the API key and password.
3. `-H` tells curl that the next bit of information sets the request header.
4. `"content-type: application/json"` lets the program we're calling (in this case, KnowledgeOwl's backend) know that the information is in JSON format.

If you're new to JSON, check out this introduction to JSON on w3schools.

5. `-X` tells curl we want to use a custom request method.
6. `GET` tells KnowledgeOwl we're making a GET request (fetching data).
7. Then we supply the endpoint we want to call, in this case: `https://app.knowledgeowl.com/api/head/tag.json`
8. `-d` tells curl that the next information contains data we want to include in the request.
9. And finally we have the data itself, in JSON format, filtering for just tags with `'status':'active'`.

Curl options are case sensitive. For example, -H sets the request headers, while -h will open the curl help page.

Calling an API: Postman

Let's look at how to make an API request with Postman.

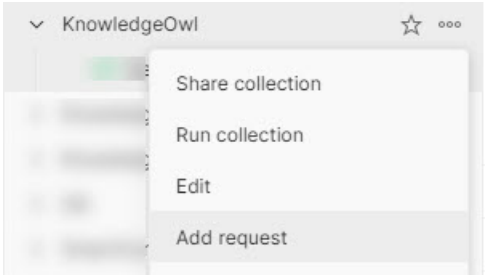
You'll need to download and install Postman, and sign up for an account. The free tier has everything needed for

this tutorial. You also need an API key. An API key is a way of authenticating API users. It makes sure that only people with the key can access the data.

We'll start by going through the steps to make a request, then take a closer look at the elements of the request.

It's a good idea to create a separate knowledge base, or at least a hidden test area in your existing one, to experiment with. The example here only retrieves information, so it shouldn't make any changes, but it's always safer to learn and experiment away from your live knowledge base.

1. Open Postman.
2. If you have used Postman before, you may be a member of several workspaces. If necessary, switch to your own workspace by selecting **Workspaces** > **My Workspace**.
3. Make sure **Collections** is selected.
4. Select **New**. Postman opens a modal allowing you to create new requests, collections, and other Postman features.
5. In the modal, select **Collection**. Postman creates a new collection.
6. Give your collection a name, for example, "KnowledgeOwl".
7. On the collection's **Authorization** tab, select **Basic Auth** as the authentication type.
8. Enter your API key in the **Username** field, and any value (for example, X) in the **Password** field. Save your changes.
9. Select the menu icon on the collection, then select **Add request**. Postman creates a new request and opens it for editing.



10. Give the request a name. We're going to get a list of all the active (available) tags in your knowledge base, so the name could be something like "Get all active tags". If your knowledge base doesn't have any tags, you'll need to create a couple to test.
11. In the **Enter request URL** field, paste in https://app.knowledgeowl.com/api/head/tag.json.
12. Select the **Body** tab.

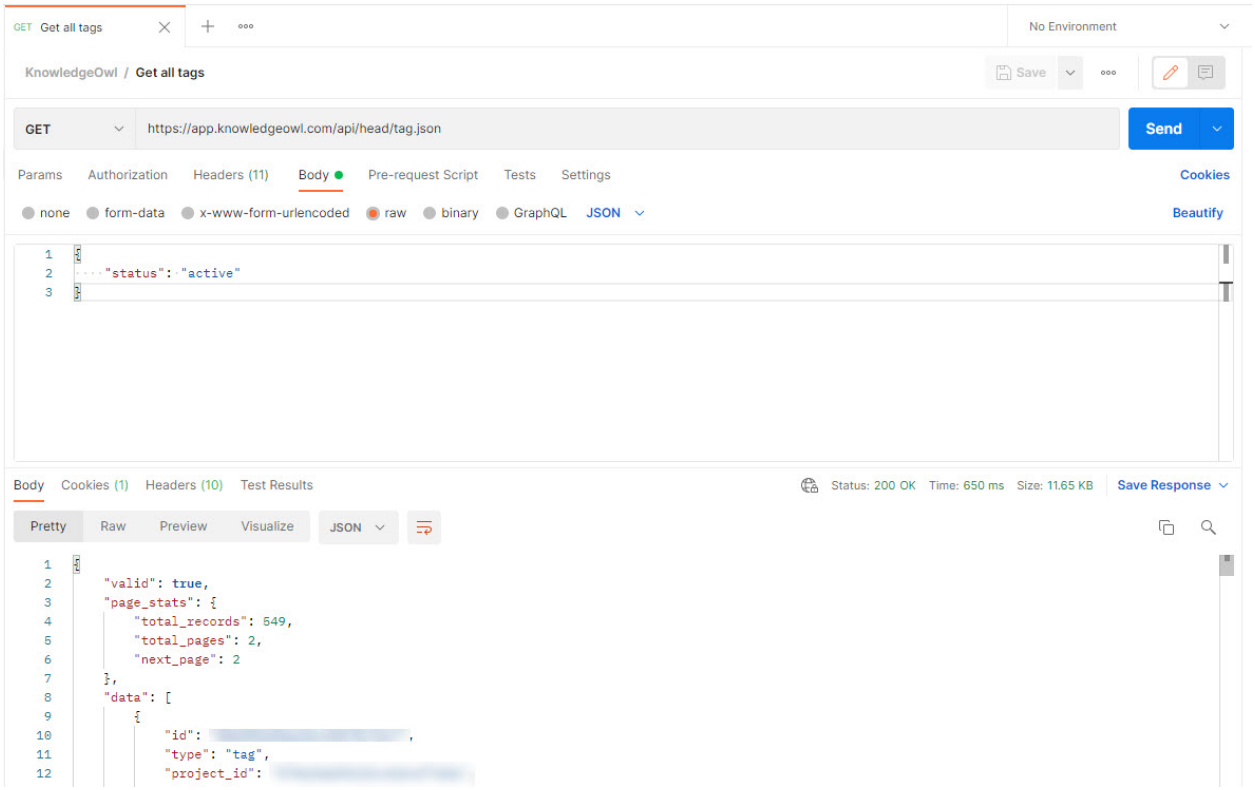
13. Set the data type: select **raw**, then a dropdown appears with **Text** in it. Change this to **JSON**.

> If you're new to JSON, check out this introduction to JSON on w3schools.

14. Enter the following in the body text box:

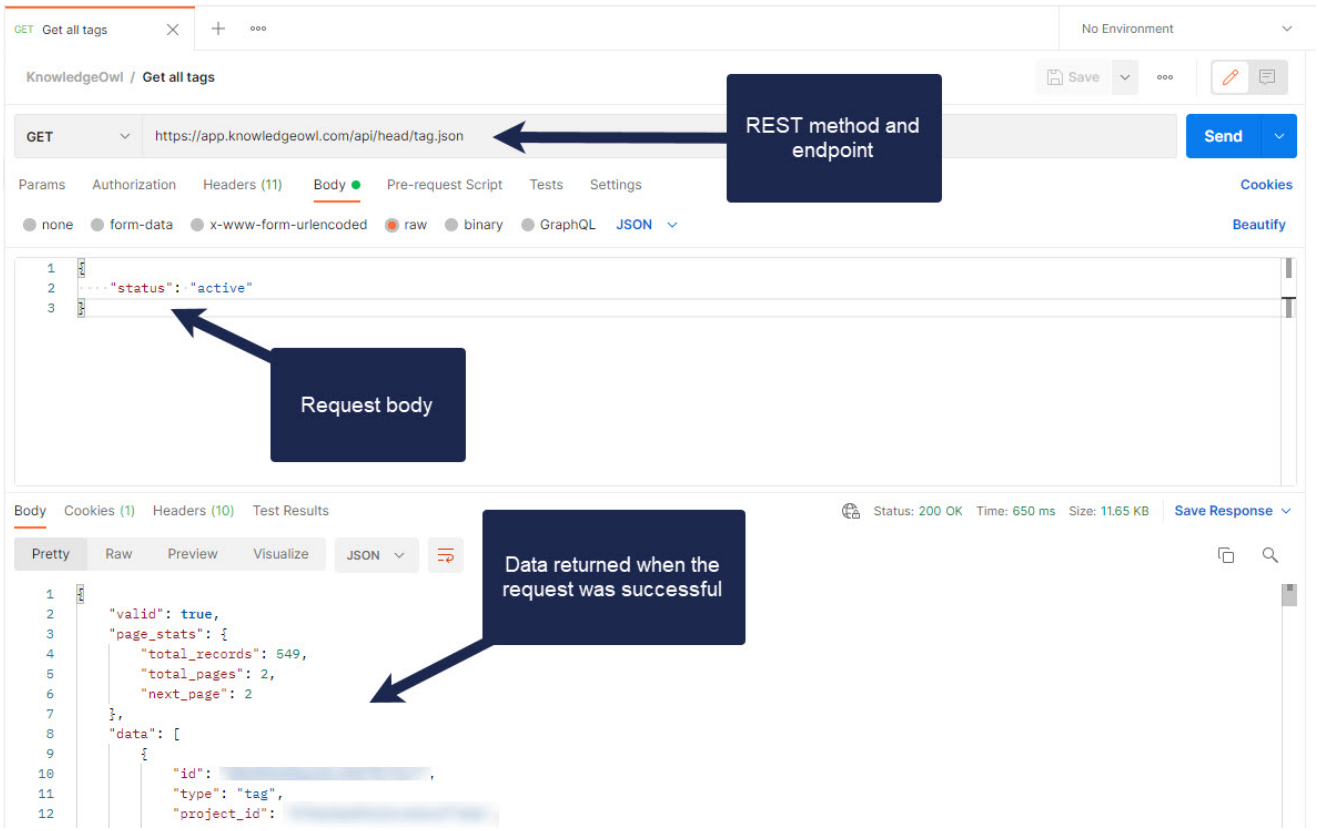
```
{ "status": "active" }
```

15. Select **Send**. Postman calls the KnowledgeOwl API, and displays the result.



Let's take a closer look at the details of the API request:

- We set up authentication on the collection. All requests within the collection inherit that setting.
- In the request itself, there are just two settings to think about for this request: the method (GET) and the endpoint URL.



Learn more

There are lots of resources out there to learn more about APIs. Here are a few of our favorites.

KnowledgeOwl API

Read more about our own API in Using the KnowledgeOwl API and our Endpoint reference.

General articles and tutorials about APIs

IBM - Application Programming Interface (API) - an overview of APIs.

Zapier - An Introduction to APIs - an introductory-level course on web APIs.

Documenting APIs

Tom Johnson - Documenting APIs: A guide for technical writers and engineers - a substantial course on REST APIs with a documentation focus. No programming required, though basic HTML, CSS, and JavaScript may be helpful.

Learn how to document your own REST API with KnowledgeOwl: REST API documentation.

JSON