# Getting Started with Postman

Postman is an API development application that facilitates the building and testing of API requests, which are important parts of implementation. This guide explains those tasks for anyone unfamiliar with Postman or request structures in general. For those familiar with the previous version of Kibo's Order Management, the authentication process is slightly different when interacting with Kibo Composable Commerce Platform APIs. These APIs use a new standard authentication mechanism through OAuth 2.0 JWT and may also require additional header parameters such as x-vol-tenant and x-vol-site values. All of these aspects are detailed in this guide.

Should an API request not be working as expected, testing it in Postman can help determine what is going wrong. Postman's interface helps generate authentication and requests to ensure that they are formatted appropriately and check responses.

## Authentication

The new OAuth 2.0 access token can be generated either with Postman's tools while defining a collection or API call, or performed as its own standalone call. In each case, two prerequisites are required to generate this token. Both of these are assigned to an implementation by Kibo and accessible in the Dev Center as shown in the below screenshot:

- Application (Client) ID
- Client Secret
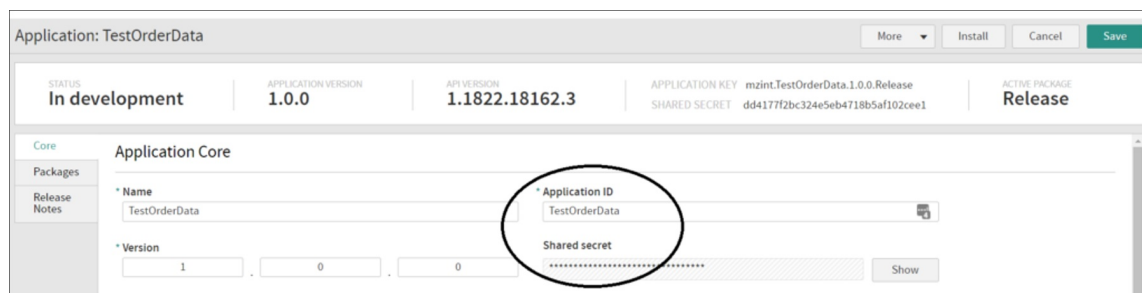


**Obtaining the Access Token via Postman Tools**

Postman will allow you to obtain a token under the Authorization tab of either an API call or at the collection configuration level.

1. In the **Authorization** tab, enter "OAuth 2.0" as the Type and "Request Headers" as the destination for auth data.
2. Click **Get New Access Token**.
3. This will open a module with fields for auth parameters.

You can generate Access Tokens with the URL `.../api/platform/applications/authtickets/oauth` . See the Creating API Requests section for more information on how to determine the base URL of this endpoint for different environments.

Once you know which environment you want to use, you can configure the module:

1. Enter one of the above URLs.
2. Provide the prerequisite A
3. pplication (Client) ID and Client Secret.
4. Click **Request Token** to generate the token.



**Obtaining the Access Token via API Call**

Instead of using the Authorization tab and subsequent tool to generate a new token, an API call can be used instead. Make a POST call to that Access Token URL endpoint, with the Content-Type set as application/json and the prerequisites in the request body as shown here.

The response will return the new Access Token.





**How to Authenticate API Requests**

The access token is always required in the header when submitting an API call, but the other required keys are not universal across all APIs. Some APIs may require only the additional x-vol-tenant key, while others will also require a x-vol-site key. The tenant and site IDs are assigned per implementation by Kibo.

This example shows what the header would look like for an API call that requires the tenant key but not the site key.



# Creating API Requests

There are four important elements to an API request:

1. URL

2. Method

3. Header

4. Body

## URL

This is the unique address or endpoint for a resource, webpage, image, video, etc. The base URL of this endpoint will depend on whether your tenant is in a sandbox or production tenant, as well as whether it is in the US or EU. These endpoints must include a tenant identifier ( `t00000` in the templates below) and any additional identifiers ( `tp0` and `euw0` ) assigned by Kibo. You can optionally include a site identifier in addition to the required tenant (which would make the Base URL `t10000-s00000` instead of `t10000` ).

- Example US Sandbox Tenant: `https://t10000.sandbox.mozu.com/api`
- Example US Production Tenant: `https://t10000.tp0.mozu.com/api`
- Example EU Sandbox Tenant: `https://t100000.sb.euw0.kibocommerce.com/api`
- Example EU Production Tenant: `https://t100000.tp0.euw1.kibocommerce.com/api`

Different sorting and filtering syntaxes can be applied to many GET calls. Do not forget that the parameters used in the URL are case sensitive.

## Method

The method is the action that the request would like the server to take (GET, POST, PUT, DELETE, etc.). The method can be selected in the dropdown next to the URL in Postman.
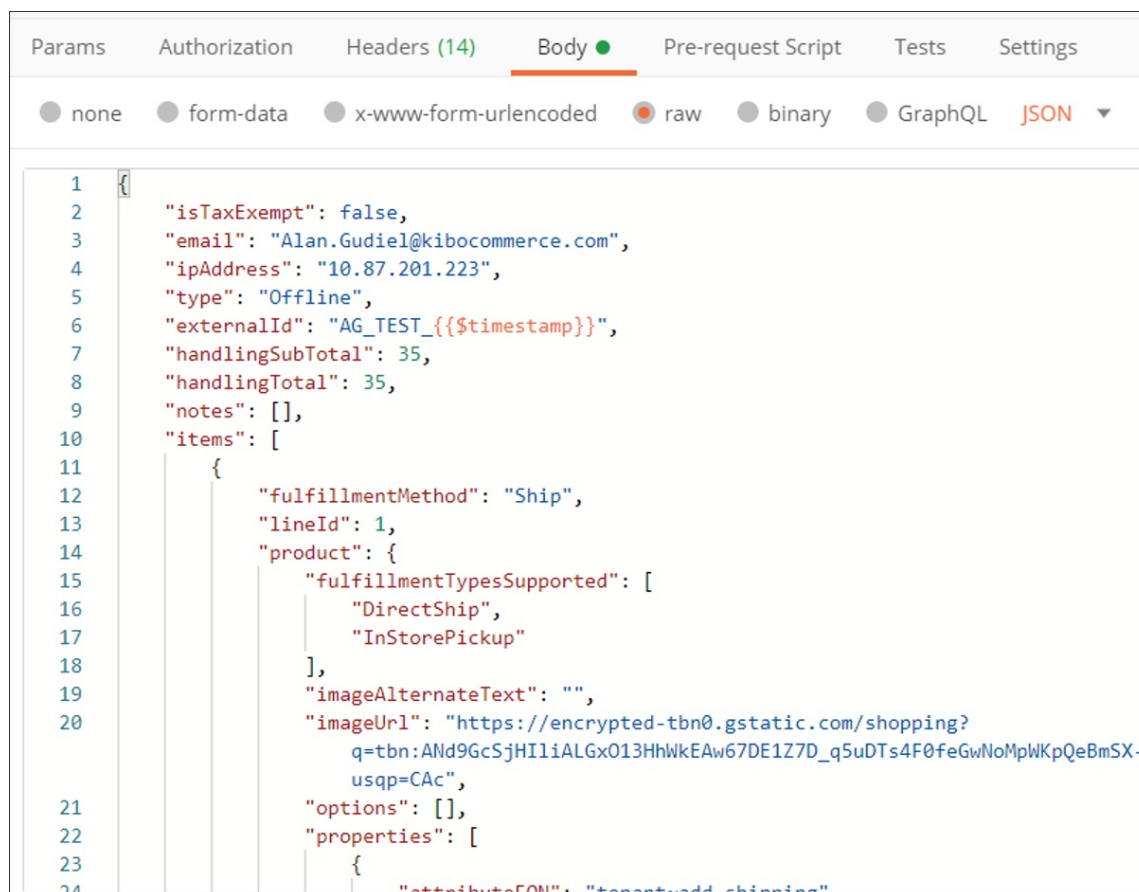
## Header

The header provides various parameters and meta-information about the request being made.



## Body

The body contains data that is being sent to the server. Some requests, such as GET, do not require any information to be passed in the body. Others, such as POST or PUT, require information for the server to be passed in the body in a certain format and with certain parameters.

# Variables and Header Presets

Variables can be added anywhere in a request (e.g. URL, header, body) and are denoted with {{ }} around the value. This can save time and enforce more consistent requests by managing values at a single point of maintenance. These variables are often applied in endpoints and headers, such as for identifiers, so that the value be applied across the board.

Add and manage variables by clicking the cog button and then Manage Environments. Then select Globals or the other appropriate environment for the variables.
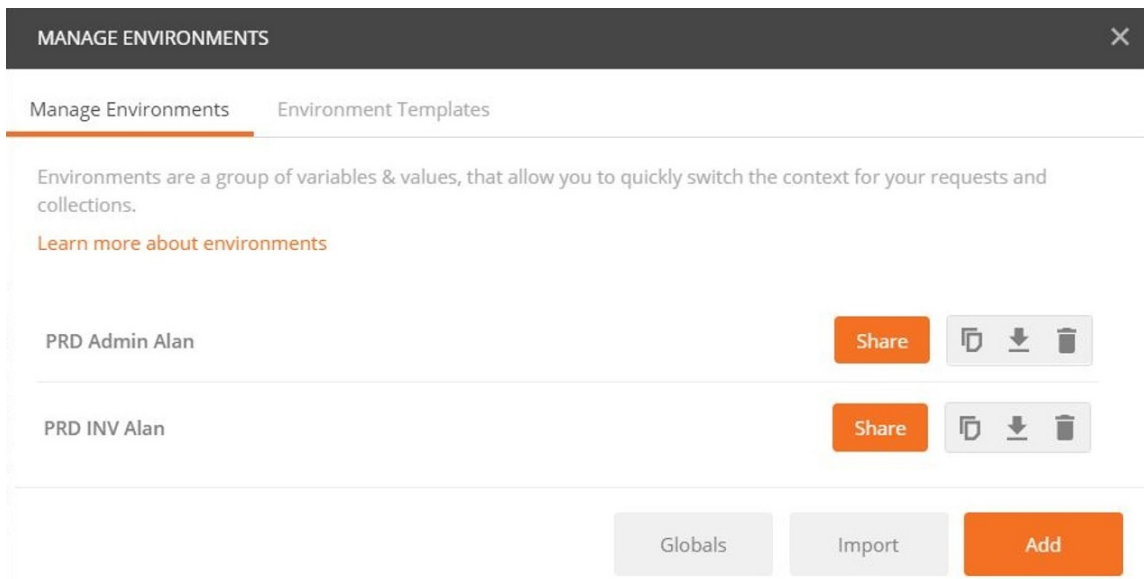
Manage and create header presets for different environments by clicking on the Presets button next to the header values. This allows further customization of header keys and values.



# Troubleshooting

Use the following tips and tricks to improve your Postman experience, or troubleshoot any errors you may encounter.

## Use the Postman Application

The web version of Postman doesn't have pre-request scripts, and that causes authentications to fail. Make sure to use the Postman application instead.

Note that while the Postman website may prompt you to sign up before downloading, you do NOT need to create an account to download the application.

## Remove New Lines in Variables

The light gray symbol shown below means that you copied and pasted a new line into your environment variable, which will cause authentication to fail. Remove this new line in order to authenticate without errors.

## Refresh Token After Adding Behaviors

If you added a new behavior, then you need to refresh your `accessToken` or `jwtAccessToken` because the token caches behaviors when it is generated. To do this, clear the `accessToken` value in your environment variables as shown below to generate a new one.



## Wait After Installing Application

It takes a few minutes after installing an application before it will be fully accepted by Reverse Proxy, so wait a short time before attempting authentication or any other action.

## Check Your Tenant URL

If you intend to work in a production tenant, you need to ensure that you are not using a sandbox base URL in your endpoint. If you attempt to use a production tenant on a sandbox URL, it will return zero results from Get Products and other APIs.

For example, if you are trying to access the t00000 production tenant:

- Bad URL: https://t00000.sandbox.mozu.com
- Good URL: https://t00000.tp3.mozu.com

## Clear Cookies if Requests Fail

If you type in the wrong URL and receive a response from the storefront, it can cause all subsequent requests to fail as well. This is because cookies take priority over the regular headers, so if you were to get assigned a guest shopper cookie by accessing the storefront then it will be used for any following requests instead of your authentication header.

The below example shows how cookies were generated from a 403 Forbidden response. To clear them and try a new API call, click **Cookies** in the top right.



## Check Media Type Encoding

If you receive a 415 Unsupported Media Type error, check that your encoding is set to "JSON" in the body of your request. If your encoding is set to another value, such as "Text" shown below, then the 415 error will be returned.



## Specify Site for Some APIs

Some APIs only make sense with a site context, while others can work based only off of the tenant. For most APIs that reference the catalog, the  x-vol-site  header must be included.
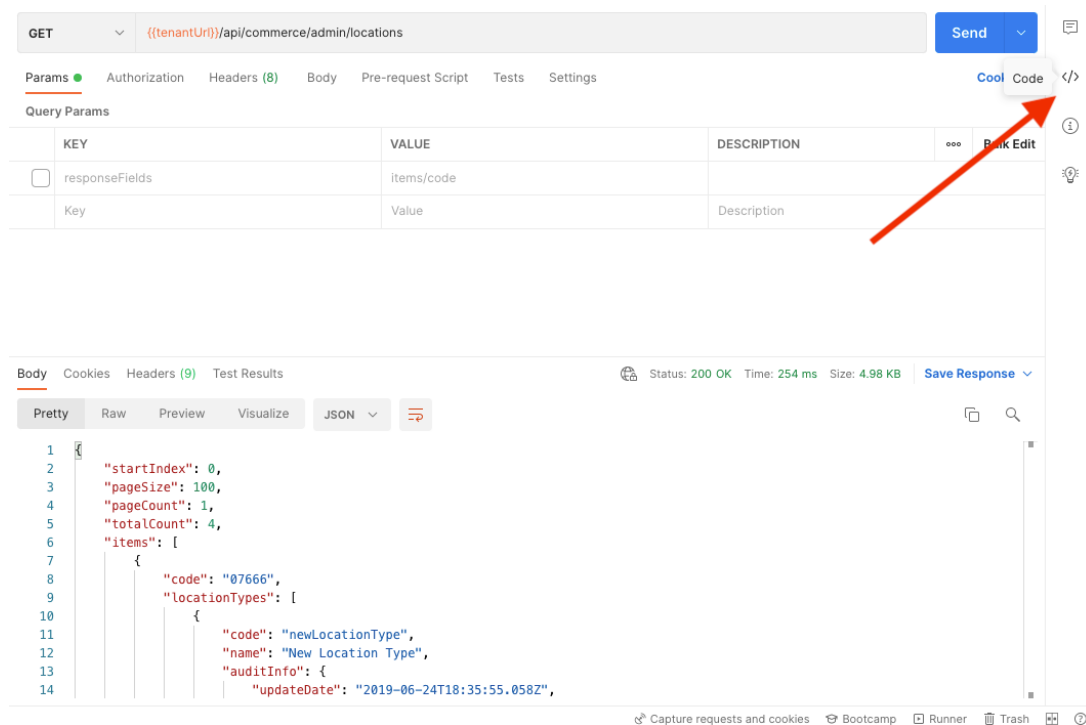
Sometimes a site can be inferred from the URL such as in `t00000-s00000.sandbox.mozu.com` . But if your URL does not include s00000 and you experience a 400 Bad Request Error or a message about missing context level, then you can try adding the `x-vol-site` header with the value of your site.

However, be careful when you use this header. It may overfilter your data if used for an API that doesn't require site context, such as the Get Shipments API. If you specified a site context for this call and there are no order on that site, you may return zero results even if there are shipments elsewhere in the tenant that you expected to receive.

## Provide cURL Request When Requesting Help

The most useful way to debug and receive help is by copying your attempted API request as a cURL request.

1. Click the **Code** button on the right side of Postman.



2. Select **cURL** for the code snippet type.

3. Copy and paste this when requesting support. It will help identify whether your URL is malformed, you're missing required headers, or your request data is incorrect.