

## Using Postman

Postman is an application that simplifies sending HTTP requests to Shopify API endpoints. It allows you to quickly send and capture requests to the Shopify API and serves as a useful tool for getting started with the Shopify API or debugging your apps. It's available as an OSX app.

Visit <https://www.getpostman.com/> to download.

---

## In this article

- [Get your Shopify API credentials](#)
  - [Working with Postman](#)
  - [About Collections](#)
  - [Where to go from here](#)
- 

## Get your Shopify API credentials

Once you have a Shopify store, you can quickly [generate your API credentials](#) using a private app.

In order to make authorized requests to the Shopify API from Postman, you will need to set your app's API key and password in the request header, as described below.



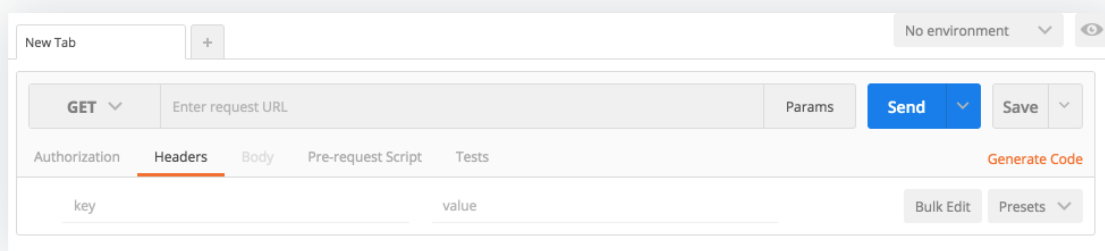
### Tip

Keep in mind that the scopes you grant here and assign to your key will give Postman the equivalent permissions. Proceed with caution, when granting

write access.

## Working with Postman

Once you're up and running with Postman, you can use tabs to make a request to the Shopify API. You can use multiple tabs to create multiple API calls quickly.



To authorize to the Shopify API:

1. Click **Authorization** and select **Basic Auth** from the **Type** dropdown.
2. For Username and Password provide your store private API key and password respectively.

### Note

Do not enter the shared secret in place of the API Password.

Once you've authenticated your request, you will need to specify the applicable HTTP method and API endpoint URL. Since you've already created a Shopify store, you can use Postman to get your store information.

To get your store information:

- Ensure that the request method dropdown menu is set to **GET** and specify the store API endpoint as follows

`https://<@store_url>/admin/shop.json`

where "*@store\_url*" is the domain of your specific Shopify store, and "*shop*" is the resource to be called.

The screenshot shows the Postman interface with the following details:

- Method: GET
- URL: `https://shopify-test-store.myshopify.com/admin/shop.json`
- Params: (empty)
- Buttons: Send, Save
- Tab: Authorization (selected)
- Other tabs: Headers, Body, Pre-request Script, Tests
- Generate Code: (link)
- Type: Basic Auth
- Buttons: Clear, Update Request
- Username: (masked)
- Password: (masked)
- Save helper data to request: (unchecked)
- Show Password: (unchecked)

- Click **Send**.

If the request was formatted correctly, the Store information is returned. To view as JSON select **Pretty** and **JSON**.

```
{
  "shop": {
    "id": 4159229,
    "name": "John's Apparel",
    "email": "johns-apparel@gmail.com",
    "domain": "johns-apparel.com",
    "created_at": "2014-03-27T17:52:33-04:00",
    "province": "Ontario",
    "country": "CA",
    "address1": "150 Elgin St.",
    "zip": "K2P1L4",

    ...

  }
}
```

## Url Format

Note that the Store url format is as follows:

`https://apikey:password@hostname/admin/resource.json`

As an alternate approach to using Postman, you can paste a properly formatted url (including API key and password) directly into the url bar. When you click **Send** the same data is returned to Postman in the Body of the response.

---

## About Collections

After you've successfully formatted an API request with Postman, you can save it into a Collection. Collections group together related API calls and are a great way to organize your work in Postman.

To create a Collection:

- Using the API call that you just sent, click the arrow next to **Save**.
- Click **Save As** and select an existing Collection or create a new Collection. You can also provide a Collection description to easily define the purpose of your Collection.

## Adding additional calls to the Collection

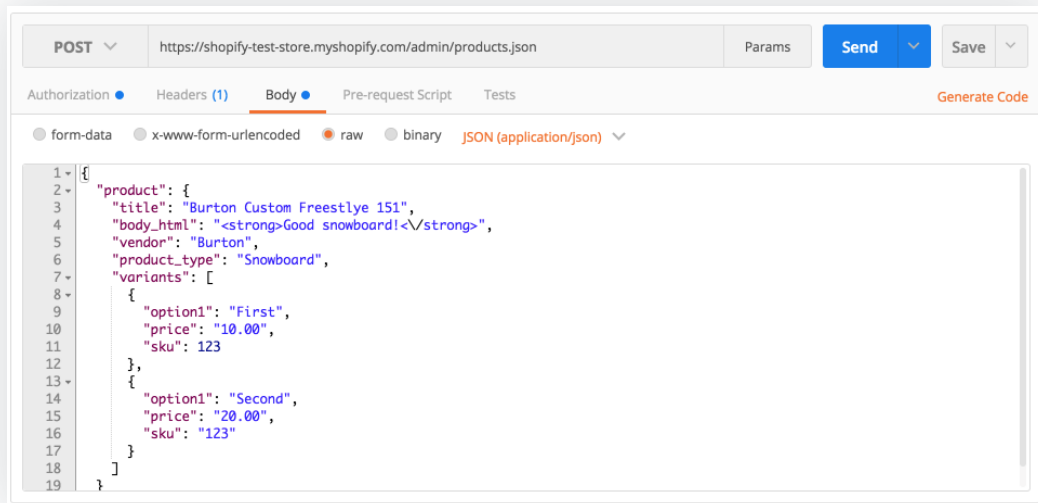
Now that you've created a Collection, you can add an additional call to help organize your Shopify API calls. For this example, you will use a call that posts some data to your Shopify store. Since you've already created a store and used Postman to view your shop information, you can use the Product endpoint to create a new product.

### Tip

Products are items for sale in your store; they can have multiple variations (variants) such as color or style, and they can have options, which are custom property names.

To create a product and add it to your Collection:

1. Open a new tab and enter the authorization information, as before.
2. Select **POST** from the API actions dropdown, and enter the endpoint url for adding a product: `https://<@store_url>/admin/products.json` (remember to replace `@store_url` with your store's actual domain).
3. Click **Body** and select "raw"; from the dropdown select "JSON (application/json)."
4. Go to the Shopify API reference and navigate to the [Product](#) endpoint.
5. Scroll down to the "Create a new product" POST call. Take note of the JSON syntax for posting a new product. For this tutorial we will copy the JSON of the Burton Custom Freestyle product.
6. Add the JSON for your product in the Body of the request.



7. Click **Send**. Provided the request is properly formatted, the product response JSON will be returned to Postman with "Status: 201 created." Also note that the Headers of the request automatically update to show the addition of "Content-Type:application/json."



8. Add the POST request to your Shopify Postman Collection, as before.

You are now up-and-running with Postman and the Shopify API. Add as many API calls to your Collection as you would like.

## Where to go from here

Browse the [Postman documentation](#) to learn about Collections and other great features in more detail. As a next step, you might also want to follow this tutorial to [build a public Shopify app](#).

This page was printed on **20 Nov 2017**. For the latest version, please go to <https://help.shopify.com/api/tutorials/using-postman/>.