POST MAN:

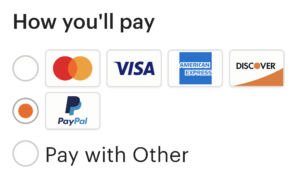
* Postman is a powerful API testing suite which has become a must-have tool for many developers & Testers.
* Used to make beautiful products to build amazing APIs and improve developer productivity.
* Used by well over a million developers around the world, and this number is rapidly growing.
* Postman is developed by a small team of talented people based out of Bangalore.
* It is a minimalist free-form guideline for defining APIs which can be easily extended for applications and is highly portable.

API:

* API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.
* API is the messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

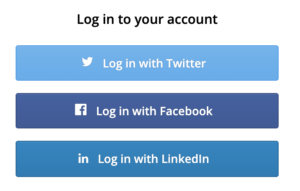
**1.Pay with PayPal**

When the user clicks the “Pay with PayPal” button, the application sends an “order” request to the PayPal API, specifying the amount owed and other important details. Then, a pop-up authenticates the user and confirms their purchase. Finally, if everything goes to plan, the API sends confirmation of payment back to the application.



### 2. Log-in Using XYZ44

Every time the application loads, it uses the API to check whether the user is already logged in by means of whatever social media platform. If not, when the user clicks the “Log-in Using XYZ” button, a pop-up opens where they are asked to confirm they actually want to log-in with that social media profile. When the user confirms, the API provides the application with identification information, so it knows who’s logging in.



Types of Api’s:

1. Open APIs: Also known as Public API, there are no restrictions to access these types of APIs because they are publicly available.
2. Partner APIs: A developer needs specific rights or licenses in order to access this type of API because they are not available to the public.
3. Internal APIs: Also known as Private APIs, only internal systems expose this type of API. These are usually designed for internal use within a company. The company uses this type of API among the different internal teams to be able to improve its products and services.
4. Composite APIs: This type of API combines different data and service APIs. It is a sequence of tasks that run synchronously as a result of the execution, and not at the request of a task. Its main uses are to speed up the process of execution and improve the performance of the listeners in the web interfaces.

# Difference between most commonly used Web Service APIs SOAP vs REST Api’s:

SOAP REST     This is a protocol that uses XML as a format to transfer data. Its main function is to define the structure of the messages and methods of communication. It also uses WSDL, or Web Services Definition Language, in a machine-readable document to publish a definition of its interface REST is not a protocol like the other web services, instead, it is a set of architectural principles. The REST service needs to have certain characteristics, including simple interfaces, which are resources identified easily within the request and manipulation of resources using the interface It has strict rules and advanced security to follow. There are loose guidelines to follow allowing developers to make recommendations easily It is driven by Function It is driven by Data It requires more Bandwidth It requires minimum Bandwidth

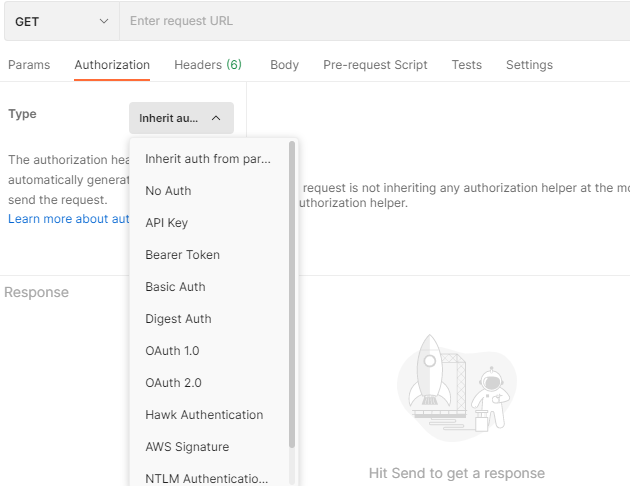
## HTTP METHODS

SL.NO TYPE DESCRIPTION 1 Inherit auth from parent This is the default auth type. In this, authorization type will be the same as the parent, which means whatever the auth type you selected for that collection (parent); the same auth type will be selected for the request, which is under that collection. 2 No Auth If you select this type, Postman will not send any auth data with the request. 3 API Key With API key auth, you send a key-value pair to the API either in the request headers or query parameters 4 Bearer Token This auth type allows the Authorization of requests by using an access key. 5 Basic Auth Removes all current representations of the target resource given by a URI 6 Digest Auth Establishes a tunnel to the server identified by a given URI 7 OAuth Describes the communication options for the target resource 8 TRACE Performs a message loop-back test along the path to the target resource

## HTTP METHODS

SL.NO TYPE DESCRIPTION 1 Inherit auth from parent This is the default auth type. In this, authorization type will be the same as the parent, which means whatever the auth type you selected for that collection (parent); the same auth type will be selected for the request, which is under that collection. 2 No Auth If you select this type, Postman will not send any auth data with the request. 3 API Key With API key auth, you send a key-value pair to the API either in the request headers or query parameters 4 Bearer Token This auth type allows the Authorization of requests by using an access key. 5 Basic Auth Removes all current representations of the target resource given by a URI 6 Digest Auth Establishes a tunnel to the server identified by a given URI 7 OAuth Describes the communication options for the target resource 8 TRACE Performs a message loop-back test along the path to the target resource

* Authorizing requests include authenticating the identity of the client who sends the request and verifies whether the client is allowed to access and conduct the endpoint operations.
* APIs use authorization details to make sure that the client requests access data safely.
* Several authorization types are available in Postman that you can use while creating an API.
* You need to ask the provider for any authorization details you require while using third party API.



# Types of Authorization

TYPE DESCRIPTION No Auth If you select this type, Postman will not send any auth data with the request. API key This is to send the Key and Value along with the API request Bearer Token This auth type allows the Authorization of requests by using an access key. Basic Auth This allows users to send username and password along with the request for API login. Digest Auth This is the two-point authentication of your API request. In this auth type user will send a request where the server will reply with a number that can only be used once. Then the user again will send the request along with username, password, and that realm number for Authorization of the request. OAuth This auth type is to access third-party API data. AWS Signature For Amazon Web Services requests, this auth type is used. Hawk Authentication This enables users to use partial cryptographic verification to authenticate their requests. Hawk Auth id, which is your API authentication ID value and Hawk Auth key, which is your API authentication key value, is required for Authorization under this auth type. NTLM Authentication (Beta) This is the auth type for windows OS and standalone systems. Akamai EdgeGrid Akamai technologies use this auth type for authorizing requests.

# Making requests in postman

You can send requests in Postman to connect to APIs you are working with. Your requests can retrieve, add, delete, and update data. Whether you are building or testing your own API, or integrating with a third-party API, you can try out your requests in Postman

Each API request uses an HTTP method. The most common methods are GET, POST, PATCH, PUT, and DELETE.

* GET methods retrieve data from an API.
* POST sends new data to an API.
* PATCH and PUT methods update existing data.
* DELETE removes existing data.

In Postman you can make API requests and examine the responses without using a terminal or writing any code. When you create a request and click Send, the API response appears inside the Postman user interface.

Understanding Get request in postman

-> Go to New

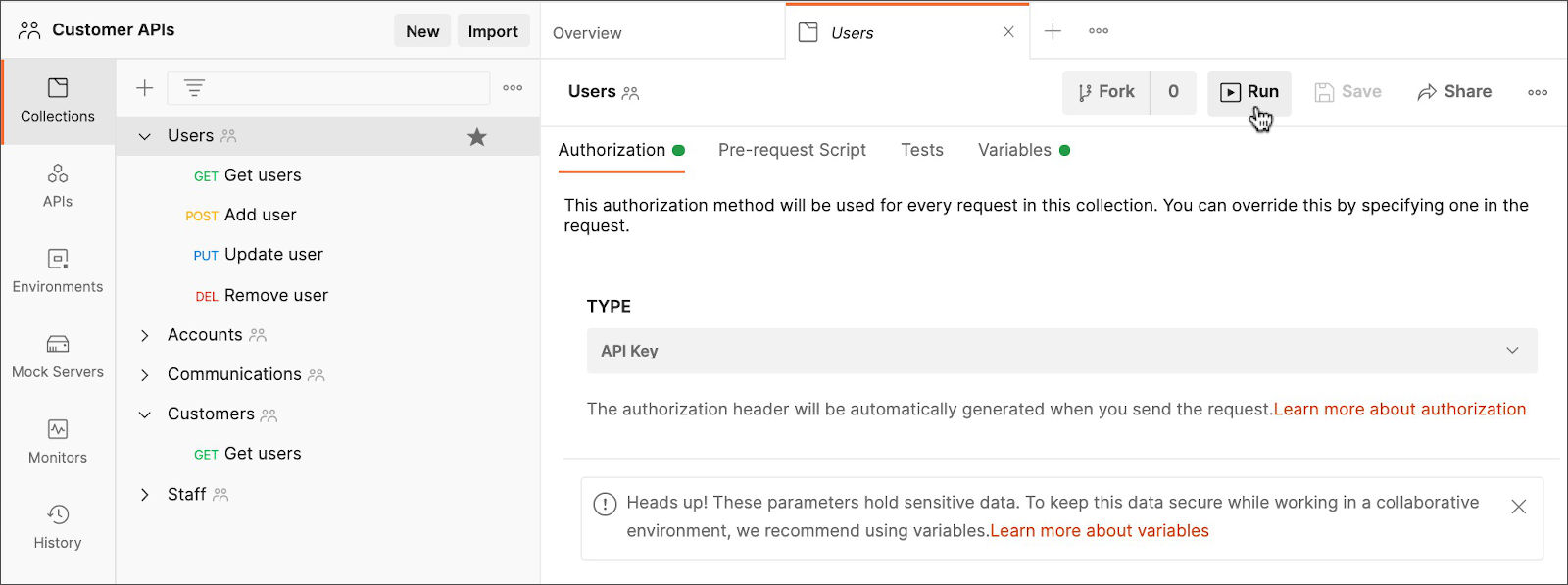
-> Create New

-> New Tab

->Select GET

-> Enter Request url

-> Send



# Understanding POST request in postman

-> Go to New

-> Create New

-> New Tab

-> Select POST

-> Enter Request url

-> Select & Enter Body

-> Send

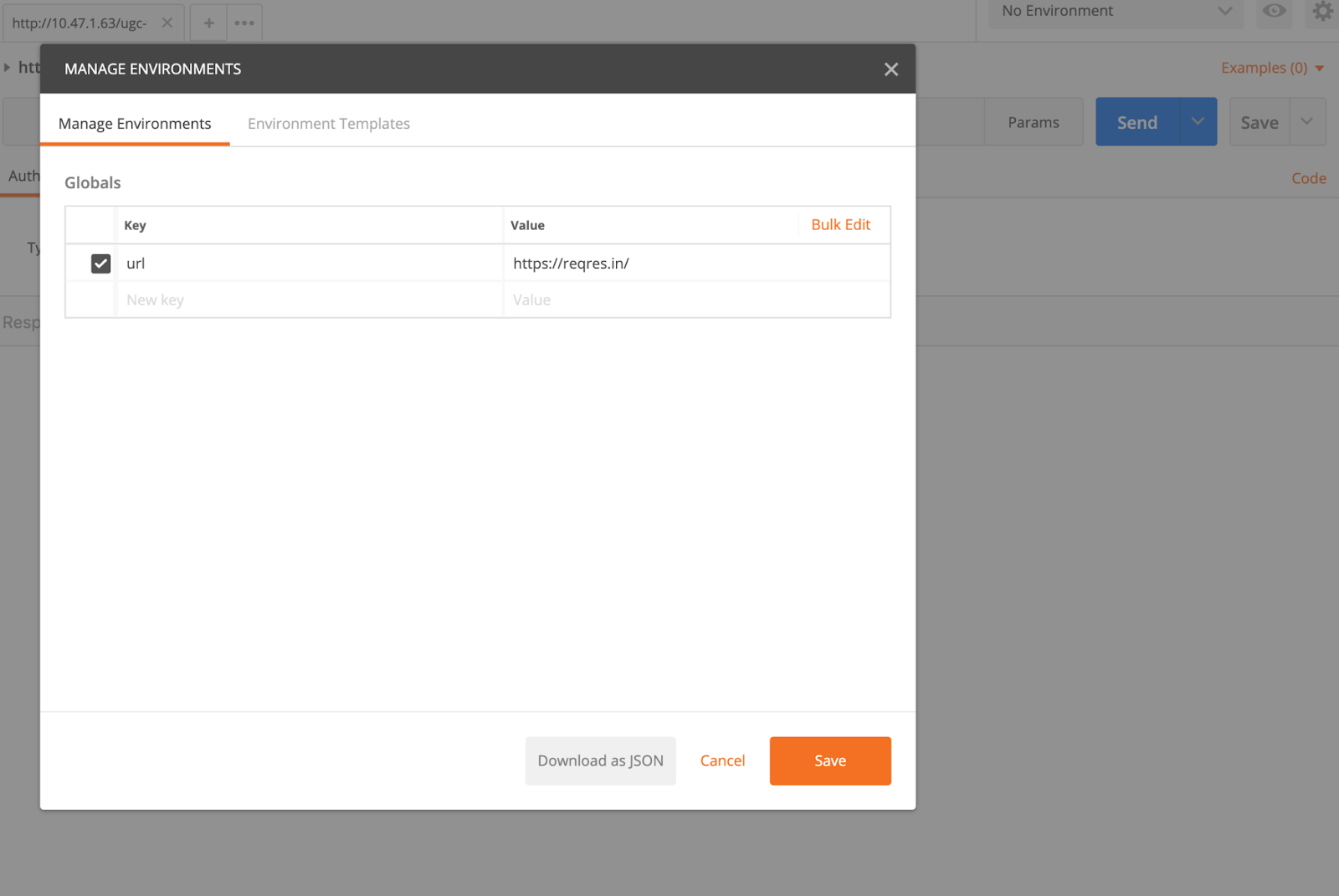
For sample post URL you can refer to this link - <https://reqres.in/>

Similarly select the respective methods for PUT and DELETE

# Creating an Environment in Postman

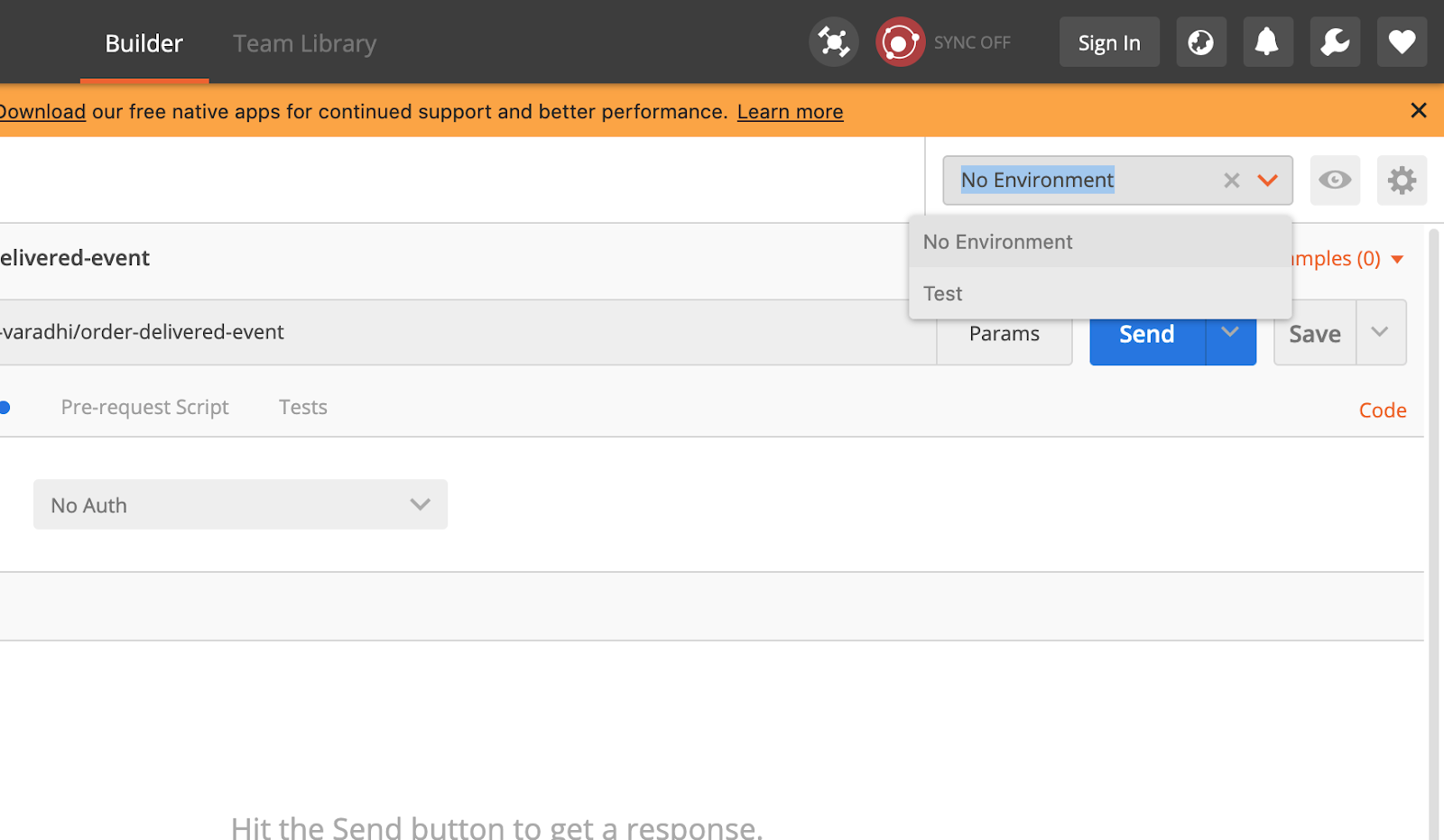
* Select the Edit option.
* Enter the below mentioned key-value pairs in which Key = url and Initial Value = <https://reqres.in/>

Then select the Save button and close the Manage Environment window. Now you have an environment variable with the name 'url', and the value of the variable is the <https://reqres.in/>



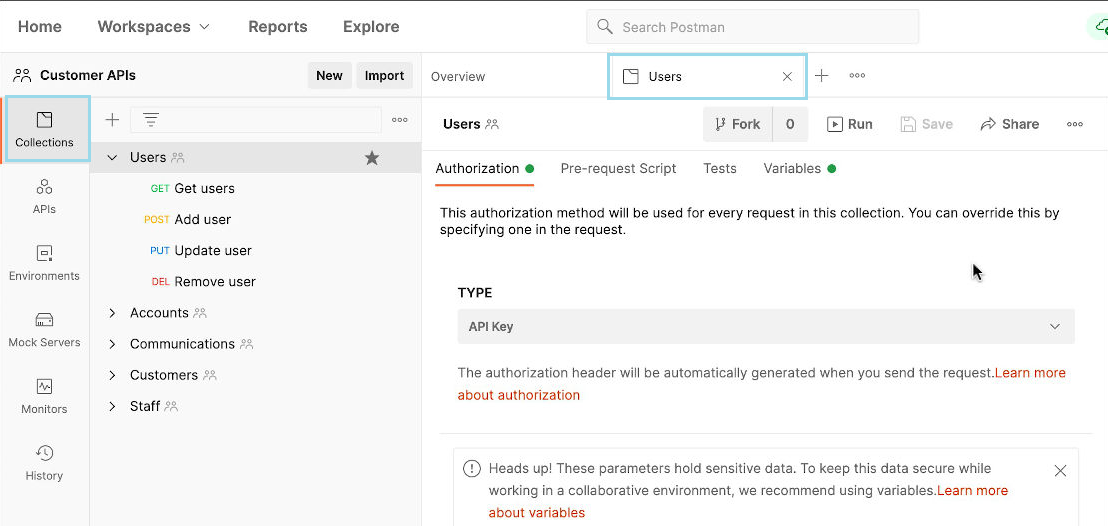
# Creating an Environment in Postman:

Select the environment from the drop-down. Now we can access the entire available variable in the environment.



You can group your Postman requests and examples into collections to keep your workspace organized, to collaborate with teammates, to generate API documentation / test suites, and to automate request runs.

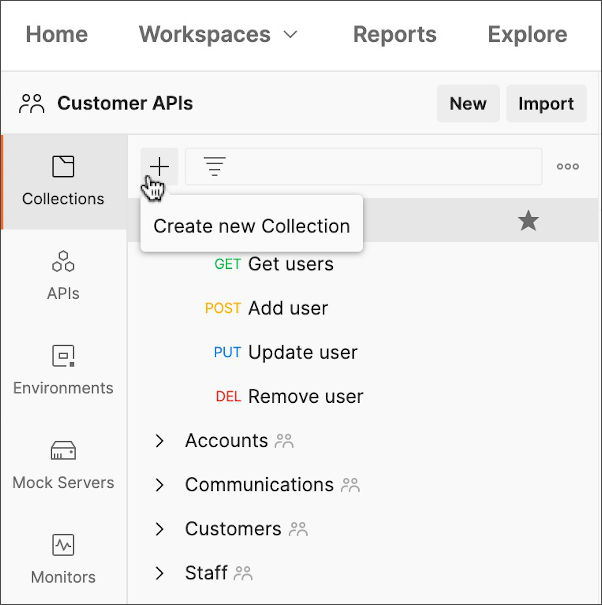
Postman lists your collections in the left sidebar > select Collections. Below is the sample image of how collections looks like in postman



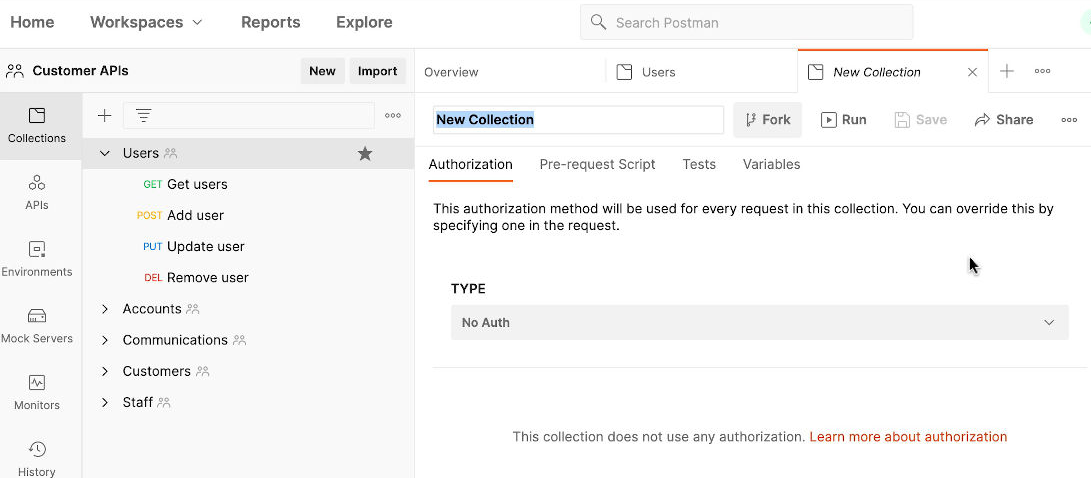
Click a collection to open its overview in a tab. You can open and close collection content by clicking the arrow to the left, and favorite (★) to move a collection to the top of the list. You can add sub-folders to create an extra level of nesting with your requests and examples. Use the search text-field to filter through your collections.

Creating collections

You can create a new collection by clicking + in the left sidebar, the New button > Collection, or the overview that appears when you first launch Postman.



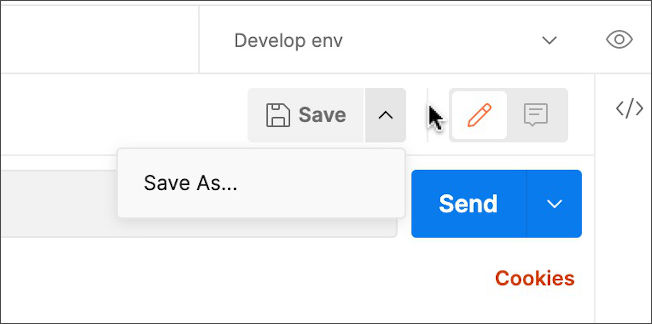
Give your new collection a name. You can start adding requests if you don't want to specify collection details until later.



# Collections

Adding requests

You can add existing requests to collections and can create new requests inside collections. If you have a request open in Postman, click Save or Save As. Choose (or create) a collection and click Save.



Managing collections

To add a folder to your collection, open its ... menu in Collections on the left of Postman, and choose Add Folder. You can then add requests and examples to the folder either by dragging them over it from inside the collection, or choosing Add a request when empty / Add Request from the ... menu.

Using collections

You can use collections to power various parts of your API development, testing, and publishing workflows. Use the buttons to the right of the collection to see documentation, comments, the changelog, pull requests, and an overview of the collection including linked elements such as mocks and monitors.

* The [Collection Runner](https://learning.postman.com/docs/running-collections/intro-to-collection-runs/) allows you to run all requests in a collection and build testing workflows into your runs.
* You can define [scripts](https://learning.postman.com/docs/writing-scripts/intro-to-scripts/) in your collection, and they will run for each request inside it.
* [Collection variables](https://learning.postman.com/docs/sending-requests/variables/) allow you to define values to use throughout the requests in the collection.
* You can generate [API documentation](https://learning.postman.com/docs/publishing-your-api/documenting-your-api/) from a collection and share it publicly as well as adding it to the [Postman API Network](https://www.postman.com/explore).
* Attaching a [monitor](https://learning.postman.com/docs/designing-and-developing-your-api/monitoring-your-api/intro-monitors/) to a collection lets you schedule collection runs.
* If you add [examples](https://learning.postman.com/docs/sending-requests/examples/) to your requests, you can use [mock servers](https://learning.postman.com/docs/designing-and-developing-your-api/mocking-data/setting-up-mock/) to return sample data during testing and development.

# Workspace

A workspace is a collaborative environment for a group of users to develop and test APIs. In simpler terms, workspace is the project in which you work. A setting done in a workspace remains in the same workspace. It is like a shell in which you work. There has to be a workspace to work in Postman

There are different types of workspace

* Team workspace

In Team workspace option, your team can test api simultaneously on one workspace and anyone can edit and update. All teammates can view and join as Admin

* Personal workspace

A personal workspace is similar to the team workspace but the difference being this workspace is completely personal to the user who created it. Only you can access

* Public workspace

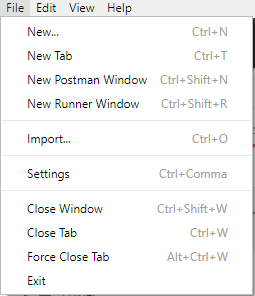
Anyone can view & all teammates can view and join as Admin

# Export & Import in Postman

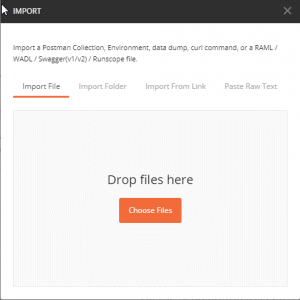
Postman can import and export Postman data, including collections, environments, data dumps, and globals. Postman can also import non-Postman data in the form of API schemas to help you consolidate your API development workflow.

How to import in Postman?

1. Click on the file tab and then click import

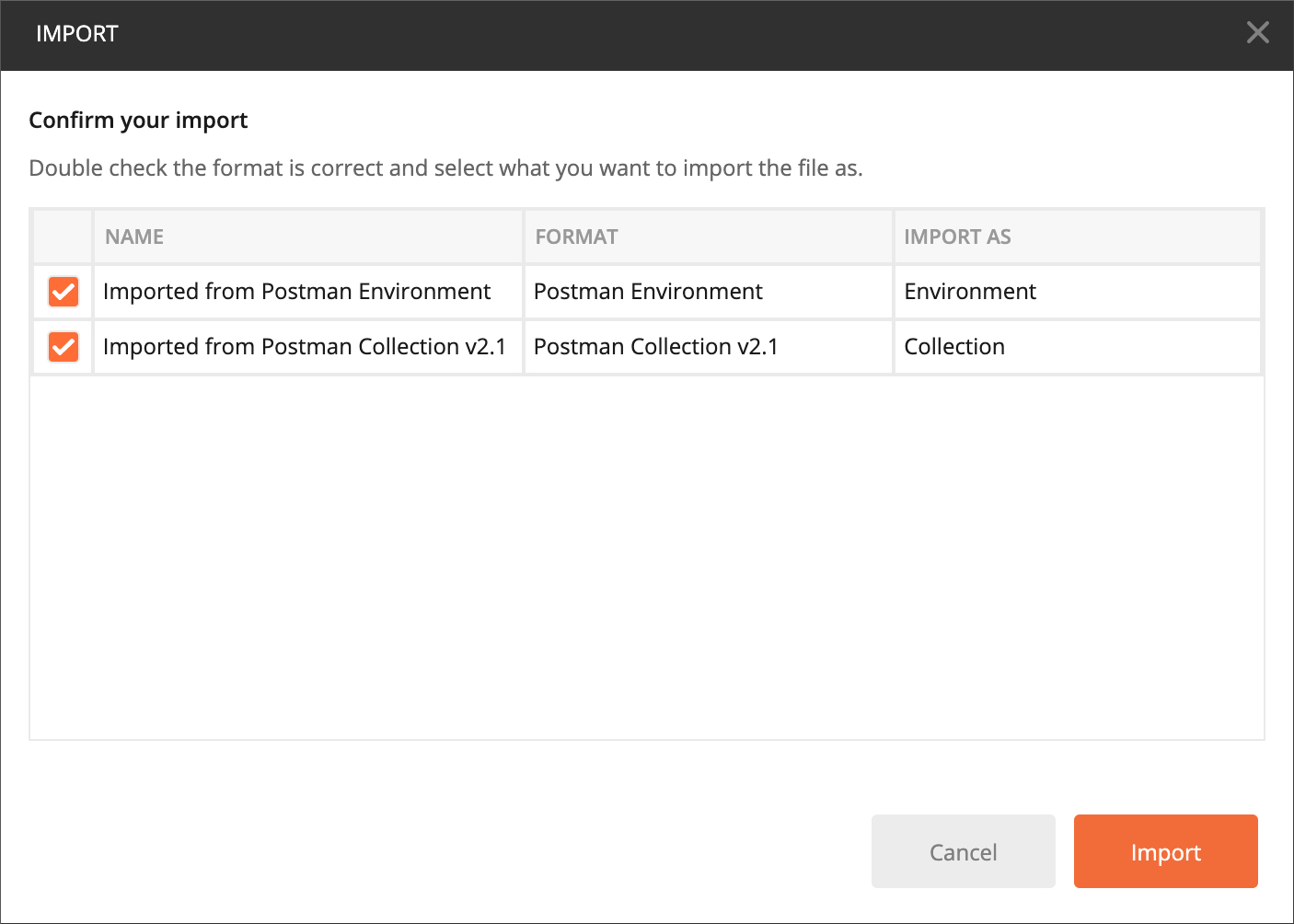


1. Choose the method you want to import an item



You can import Postman data like collections, environments, data dumps, and globals.

3. To import Postman data, click Import. Select your file or folder, input your link, paste your raw text, or [import from GitHub](https://learning.postman.com/docs/getting-started/importing-and-exporting-data/#importing-github-repositories). Postman will automatically recognize Postman data, confirming the name, format, and what the file will import as. Click Import to bring your data into Postman.



Postman lists your collections in the left sidebar > select Collections. Below is the sample image of how collections looks like in postman

