1. **POSTMAN BASICS**
2. **Introduction**()

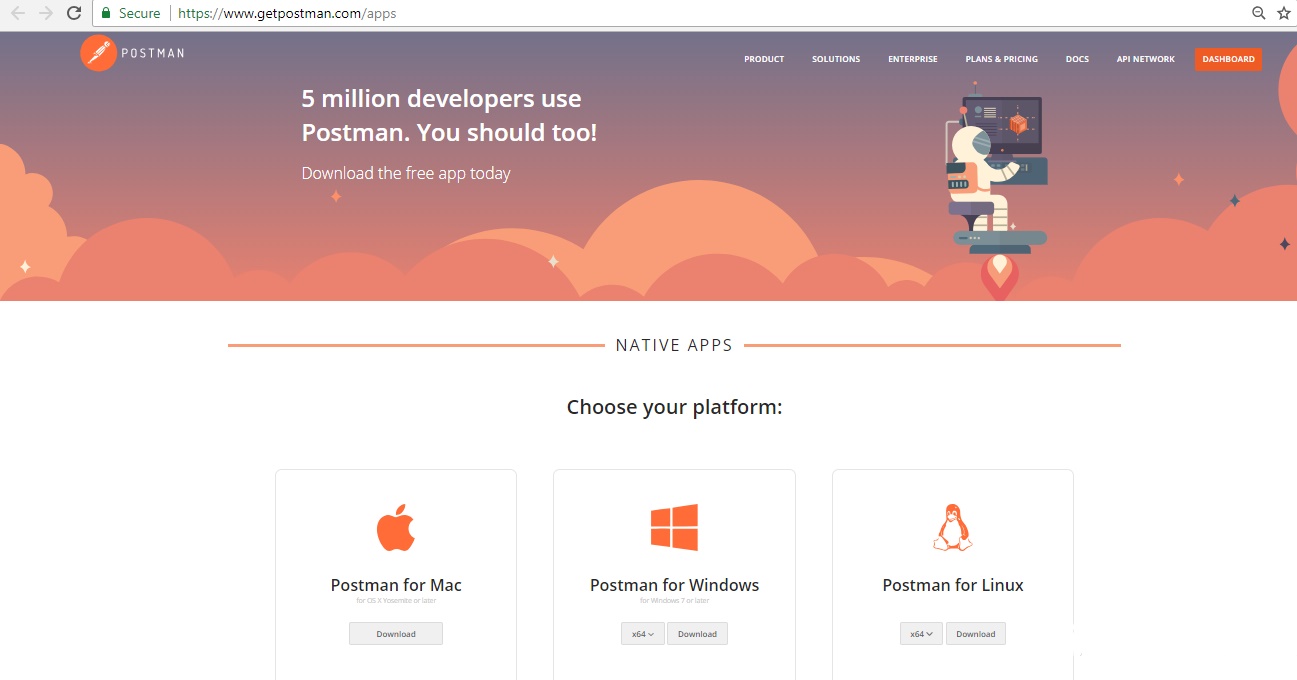
**Postman** is a powerful HTTP client for testing web services. Created by Abhinav Asthana, a programmer and designer based in Bangalore, India, **Postman** makes it easy to test, develop and document APIs by allowing users to quickly put together both simple and complex HTTP requests. **Postman** is a great tool when trying to dissect RESTful APIs made by others or test ones you have made yourself. It offers a sleek user interface with which to make HTML requests, without the hassle of writing a bunch of code just to test an API's functionality.

Speaking through the personal experience, Postman can be called as a very useful tool for effectively and efficiently testing the web api’s in no time. Testers/programmer can totally rely on this tool for the regression testing if used and maintained in a correct manner, everytime the API gets deployed. It saves so much of time and creates a great platform to test a multiple cases on a single api.

**1.1 Installation of the Postman app**

Postman is available as a native app for Mac, Windows, and Linux operating systems.

To install Postman, go to the apps page and click Download for Mac / Windows / Linux depending on your platform.

****

We recommend using the Postman native apps, but Postman is also available as a Chrome app which can only run on the Chrome browser.

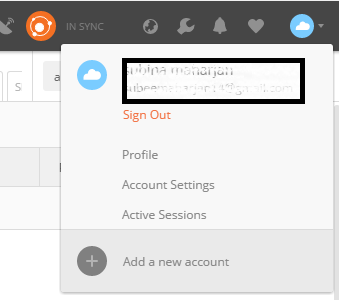
**1.2 Sign Up using Postman App**

After you have downloaded the Postman app, launch the app and you can see a prompt to log in or sign up.

Sign up with your email address or your Google account.

When you first open the Postman app, you can sign in as an existing user or create an account. If you bypass the initial sign-in option, you can click the Sign In button on the top right corner of your screen at any time.

You You can also use multiple login through **Add a new account** button at the top right corner of the screen.



Once you have successfully signed up with a gmail account, now all you have to understand is about the Postman Components. Before that, You need to know about the Syncing feature of the Postman app. **Syncing** is the feature that makes your all the data and components available on any device when you are signed in with the same postman account.

Following components are the important components of the postman.

1. **Postman Collection and Folders**
2. **Postman Requests and Responses**
3. **Postman Variables**

Let’s go through these components and its detail one by one.

1. **Postman Collection and Folders**

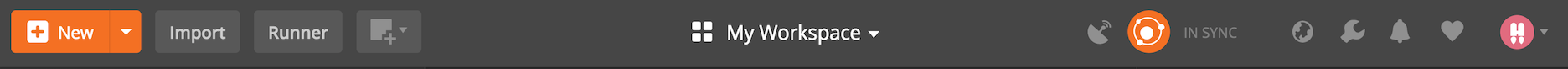
Postman Collections are a group of saved requests you can organize into folders.

Every request you send in Postman appears under the History tab of the sidebar. On a small scale, reusing requests through the history section is convenient. However, as your Postman usage scales, it can be time consuming to find a particular request in your history. Instead of combing through your history section, you can save all your requests as a group for easier access.

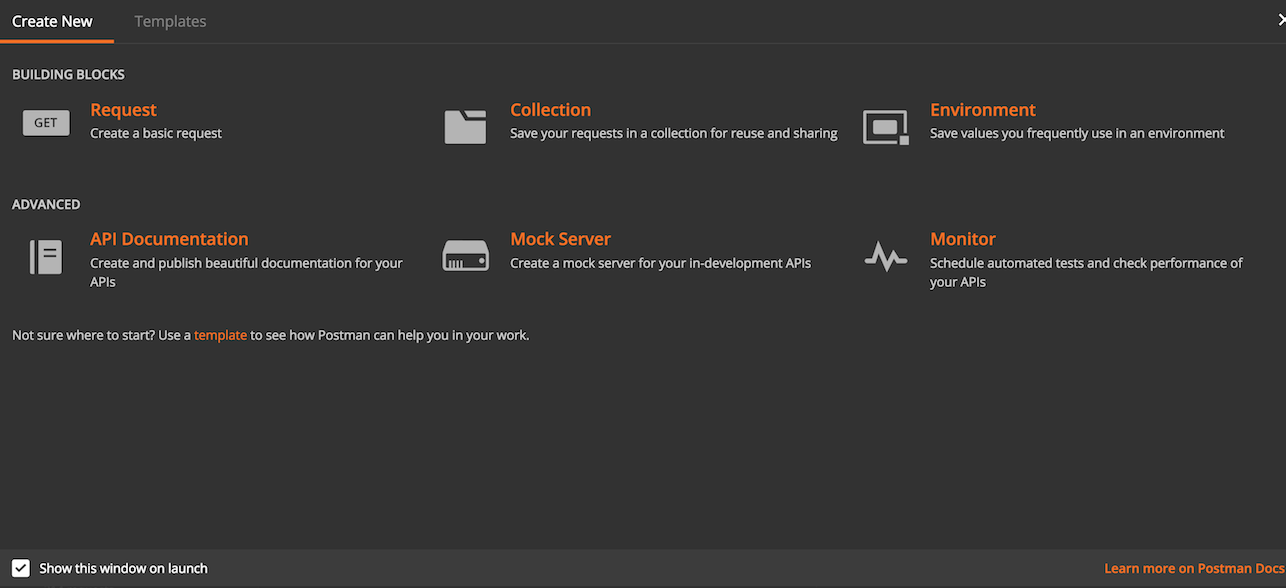
Creating workflows is important for a good organization, documentation, test suites and conditional workflows.

**How to Create a Collection:**

In the header toolbar, click the New button.

****

The Create New tab appears.

****

In the Create New tab, click “Collection” to create new Collection and click “Request” to create new Request.

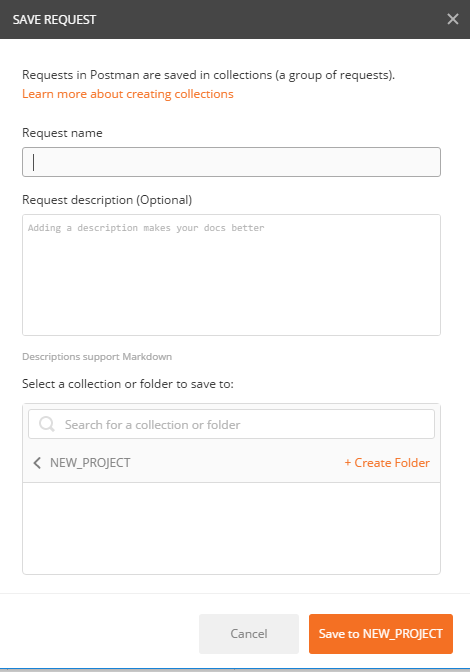
In the **CREATE A NEW COLLECTION** modal:

* Enter a name and optional description.
* Select an authorization type.

Click the **Create** button.

After creating the collection, you can save requests to the collection and add folders for better organization.

After your collection has been created, It will give you an option to add request right below the collection you have just created. So just Click **Add requests** link to create a new request.



Add a request name as you would like and select a collection or folder to save to and click **Save to Project\_Name** button. Description field is optional.

Before going inside the how to send a request, It is important to know about its few components.

* URL FIELD
* REQUEST METHODS
* AUTHORIZATION
* HEADER
* BODY
* PRE-REQUEST SCRIPT
* TESTS

**However, the four parts of an HTTP request are the URL, method, headers, and the body.**

**URL**: You need to enter the URL of the api you want to send request to, inside the URL field.

After the URL, you need to specify the **Request methods** for that URL.There could be multiple Request methods but one API request needs only one request method.Lets know some more about request methods.

* **GET**

GET is one of the most common HTTP methods.

GET is used to request data from a specified resource.

* **POST**

The data sent to the server with POST is stored in the request body of the HTTP request.

* **PUT**

The difference between POST and PUT is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly make have side effects of creating the same resource multiple times.

* **DELETE**

The DELETE method deletes the specified resource.

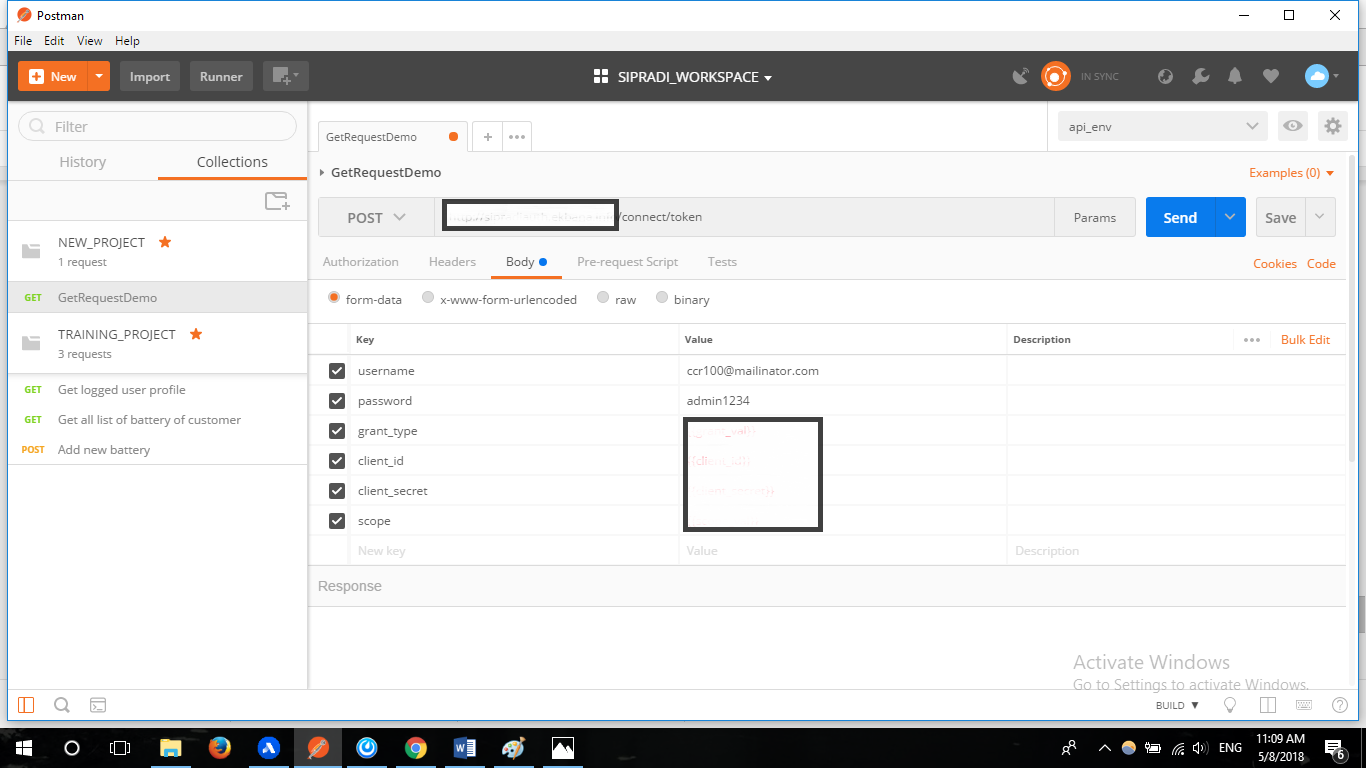
So these shall be enough for the Request method concepts for you.

**HEADERS**

You can add the Key to send to your api request under the **Headers** Presets.Commonly used Presets is the Authorization.Other could be **Grant-type**, **Password** and **Username** as per the api’s requirement.

**BODY**: While constructing requests, you’ll work frequently with the request body editor. Postman lets you send almost any kind of HTTP request. The body editor is divided into 4 areas and has different controls, depending on the body type which are form-data, urlencoded, raw and binary. But going to focus more on Form data and Raw panel as We will be using them more often.

**Form data:** Form-data is the default encoding a web form uses to transfer data. This simulates filling a form on a website, and submitting it. The form-data editor lets you set key-value pairs.



This is the example of key values entered inside the form data. These datas could also be inserted through the Bulk Edit option or one by one from the displayed field.

**RAW**:A raw request can contain anything. Postman doesn’t touch the string entered in the raw editor except replacing [environment variables](https://www.getpostman.com/docs/v6/postman/environments_and_globals/variables). Whatever you put in the text area gets sent with the request. The raw editor lets you set the formatting type along with the correct header that you should send with the raw body. You can set the Content-Type header manually too and this will override the Postman defined setting. Selecting XML/JSON in the editor type enables syntax highlighting for your request body and also sets the Content-Type header.

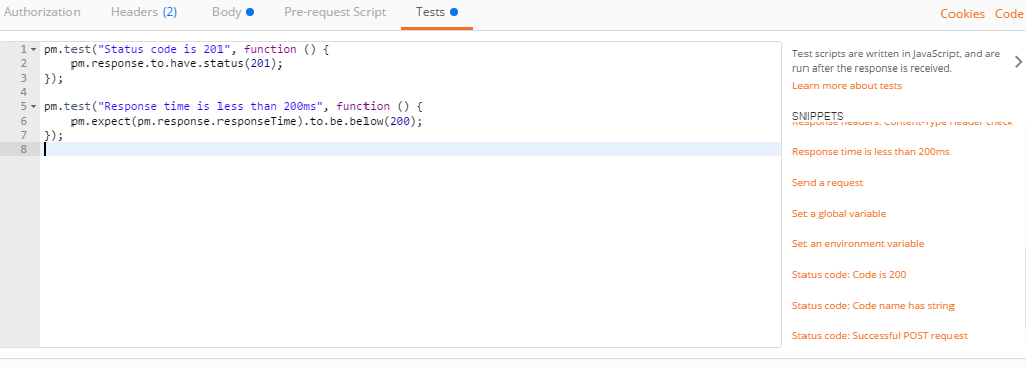


Pre-request Script

Pre-request scripts are snippets of code associated with a collection request that are executed before the request is sent.

Test Scripts: A Postman test is essentially JavaScript code executed after the request is sent, allowing access to the pm.response object. You can add the readymade js codes of the snippets available at the right side of the **test** board or add add your own js codes to make various tests.

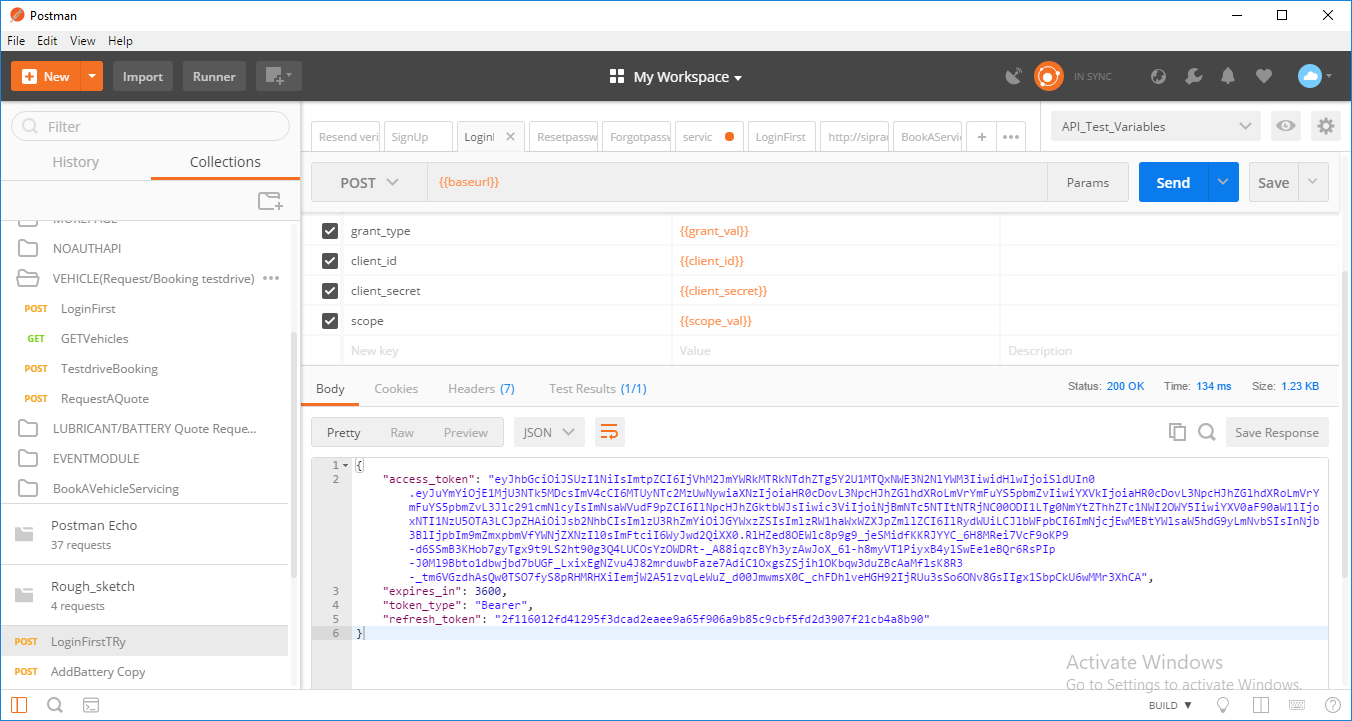
A simple example of Test Scripts is shown below.



**TEST RESPONSES:**

Test response also has few fields/components for you to understand about:

* Body
* Headers
* Test Results
* Status
* Time
* Size

****

The **body** field contains all the content provided by the Api in response.The specified Request in the image has requested for the access token in the result, hence has been provided with it.

**Headers** are displayed as key-value pairs under the **Headers** tab. Hovering over the header name can give you a description of the header according to the HTTP spec. If you are sending a HEAD request, Postman will show the headers tab by default.

**Test Result** tab shows the number of passed or failed results as per the test scripts(js scripts) you have used before while sending the Request inside the Test box.

**Status code** Status code in the response menu informs about the status of the response body. Different type of response code has a different number like that 20. You just need to know the meaning of those codes.

Few of those response codes are listed below.

200(OK)

201(CREATED)

202(ACCEPTED)

204(NO CONTENT)

302(FOUND)

400(BAD REQUEST)

401(UNAUTHORIZED)

403(FORBIDDEN)

404(NOT FOUND)

405(METHOD NOT ALLOWED)

412(PRECONDITION FAILED)

415(UNSUPPORTED MEDIA TYPE)

500(INTERNAL SERVER ERROR)

Through these status codes of these response, you can know the condition of the response body.

**Response time** Postman automatically calculates the time it took for the response to arrive from the server. This is useful for some preliminary testing for performance.

**Response size** Postman breaks down the response size into body and headers. The response sizes are approximate.

**POSTMAN VARIABLES:**

Variables allow you to reuse values in multiple places so you can keep your code DRY (Don’t Repeat Yourself). Also, if you want to change the value, you can change the variable once with the impact cascading through the rest of your code.

The following scopes are available to you:

1. Global
2. Collection
3. Environment

1. Global Variable

Global Variables are the one which can be accessed by every collection available in the [Workspaces](https://www.getpostman.com/docs/v6/postman/workspaces/intro_to_workspaces). You can set global variables for such kind of data which is required in almost every collection and is not frequently changing.

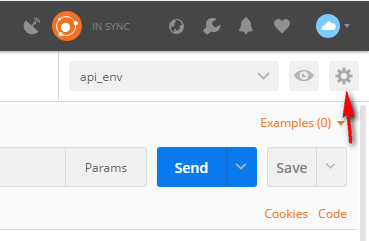
2. Environment Variable

An **environment** is a set of key-value pairs. The key represents the name of the **variable**. ... Environments let you customize requests using **variables** so you can easily switch between different setups without changing your requests. You won't have to remember all those values once they are in **Postman**.Environment Variable are the variables saved inside one Environment folder.

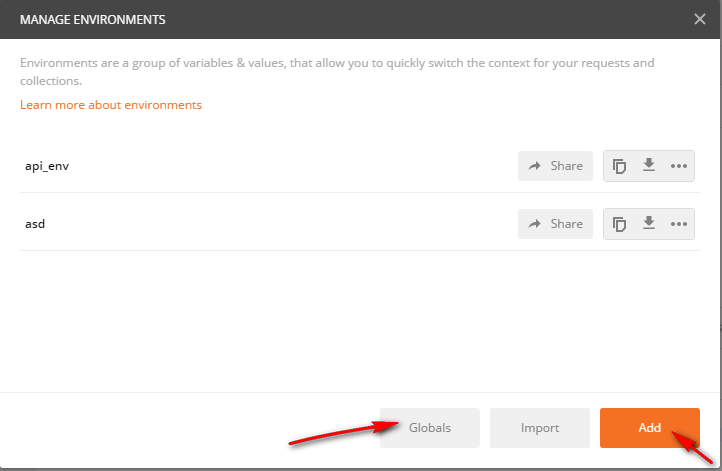
One Environment can be used for multiple collections and Multiple Environment can also be used for A single collection. You just need to make sure that you have selected the Environment which has values you require.

**How to create Global and Environment variables:**

Click on Manage Environment Icon on the top right of the screen.

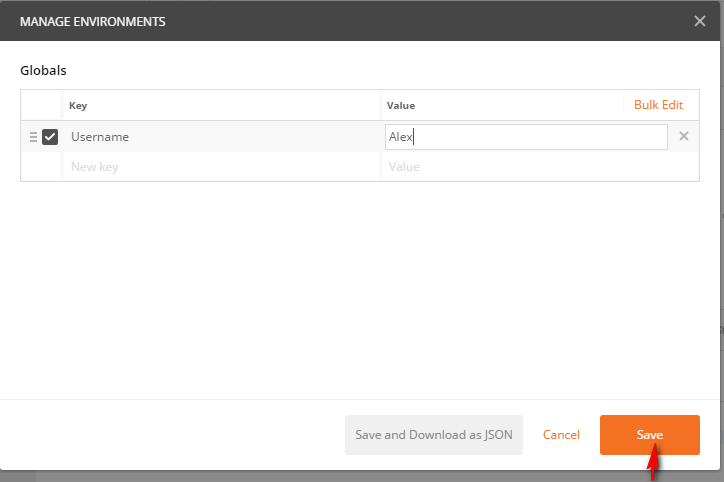


Once you Click this icon, You will see a form to add environment or global variables like below.



In this form, the right pointed **Add** button allows you to add environment variables and Left pointed **Global** Button allows you to add global variables.

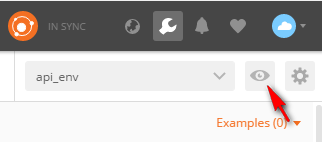
Any two button will give you a doubled columned box to add variables and its values.The box to fill up the key and values will look like below.



So in the Box like this, Add a meaningful key to input the values into, and also insert the value on the second column too.

After adding all the required number of variables, click **Save**. Now you can access those variables in the body field, URL field,Authorization field and the Test field either.

You can visualize the saved Global and environment variable from the **Environment quick look** option available at the right side of the manage environment option at the top right corner of the page.



This quick look icon will show you all the global and environment variables that you have defined.

After this, we have one more variable, Collection variable which is the variable defined inside a specific collection, only for that collection to use.The collection variable can be defined at the time of creation of the collection.If not, then it can be done through the right arrow icon available on every collection available.

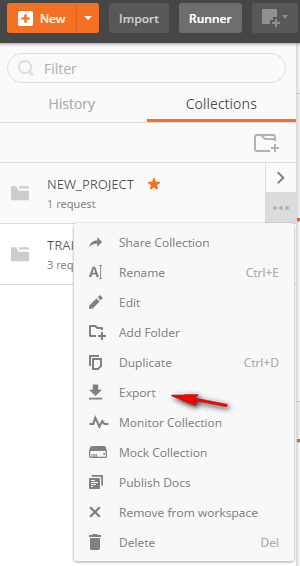
For detail information about the [variables](https://www.getpostman.com/docs/v6/postman/environments_and_globals/variables) , you can also go through the postman documentation.

**IMPORTING AND EXPORTING COLLECTION AND ENVIRONMENT**

As we already know that the collection and all the tests we have created in the postman could be synced into different devices by using the same Email. And those collections also could be shared to the team by using the premium version of postman. But what we we can do in this version to share the work we have created is to export and import the collection file or environment variable files . This also gives the way of sharing in some other way. Once the collection has been exported as a file, it could be used for desired number of times by desired number of user by using any email id simply by importing those files and folders into their postman app.

**Exporting the collection:**

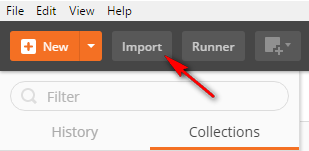
To export a collection, click the export option that appears when you click the right icon over the collection as shown in the image below.



You can then save the generated file wherever you want.

**Importing the collection:**

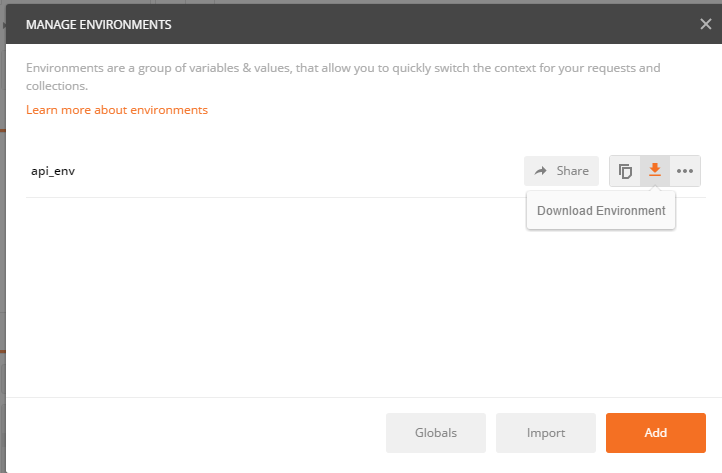
To import a collection, click the import option that is available on the task bar as shown in the image below.



You can Import through the folder, file or event the link.

**Exporting the Environment:**

To import the Environment variable, just click the download option available right beside the environment inside the manage environment dialog as shown below.



**Importing the Environment:**

The import option is available on the same dialog which is used for Exporting the environment variable as “**Import**” button.