

CamPay provides a public payment API for services such as Orange Money (OM) and Mobile Money (MoMo).

You can access CamPay API documentation from [here](#). This gives you an outline of the different endpoints, their uses, expected responses and the meaning of the different error codes.

## Authentication

As explained in class, REST is stateless. This means that the server doesn't keep track of previous communications with the client (i.e. no session).

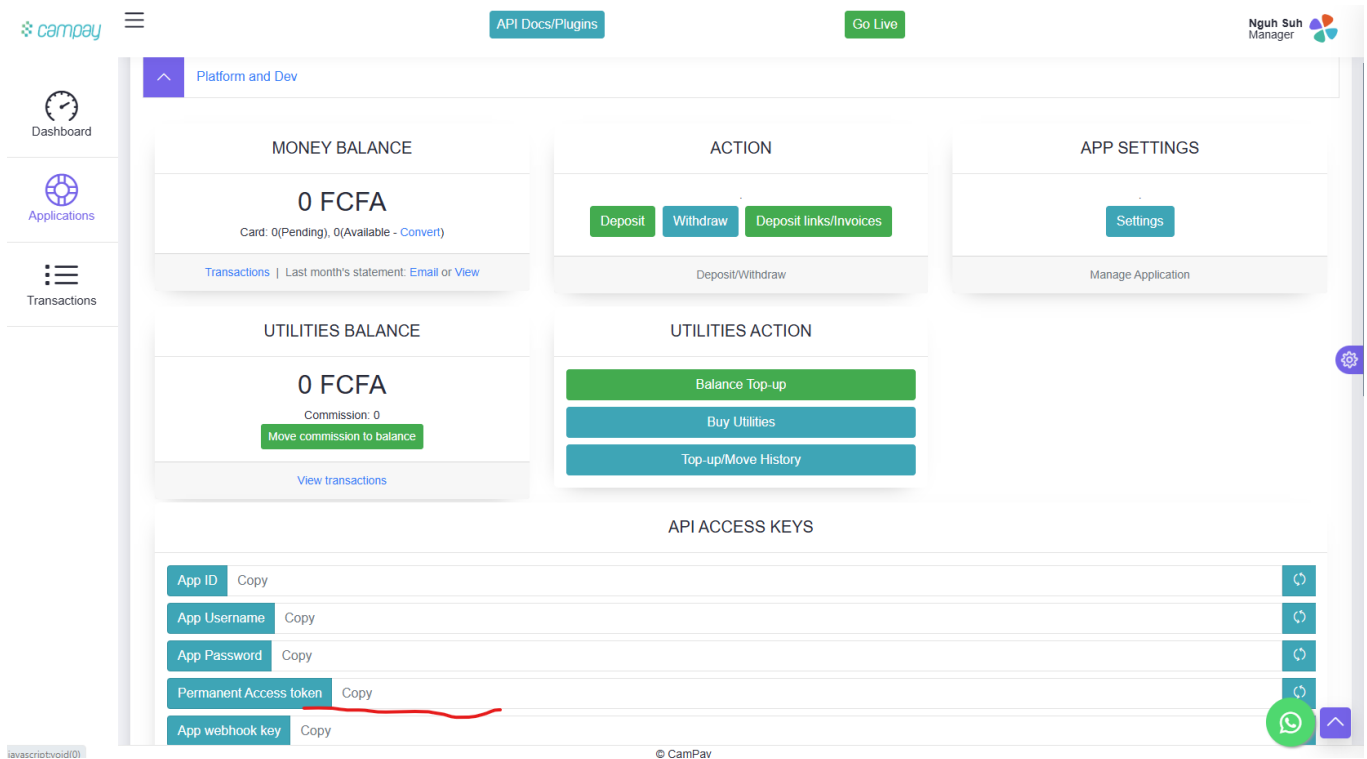
Each client request has to contain all the necessary information to complete it. Each request in a RESTful architecture should include authentication information if the user has to be authenticated.

CamPay provides 2 API authentication tokens: a permanent and temporary (expires after a given amount of time) token. For security purposes, you might want to use a temporary token but for this class, we will be using a permanent token.

## Getting the permanent token

The permanent token for the app can be found on the CamPay demo site in the "Applications Page".

Go to that page and click on the desired application, this will expand it to display some data for that application.

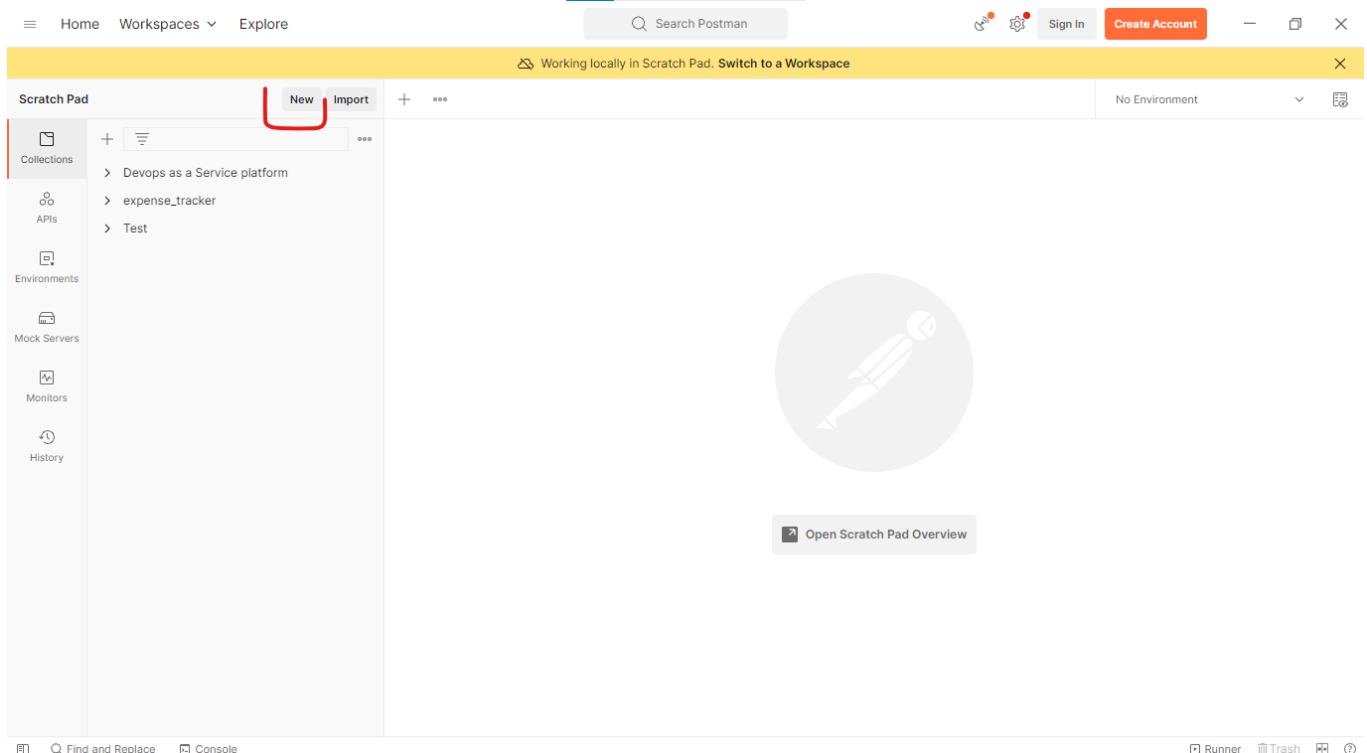


Clicking the "Copy" input field next to the "Permanent Access Token" will copy the Permanent Access Token to your clipboard.

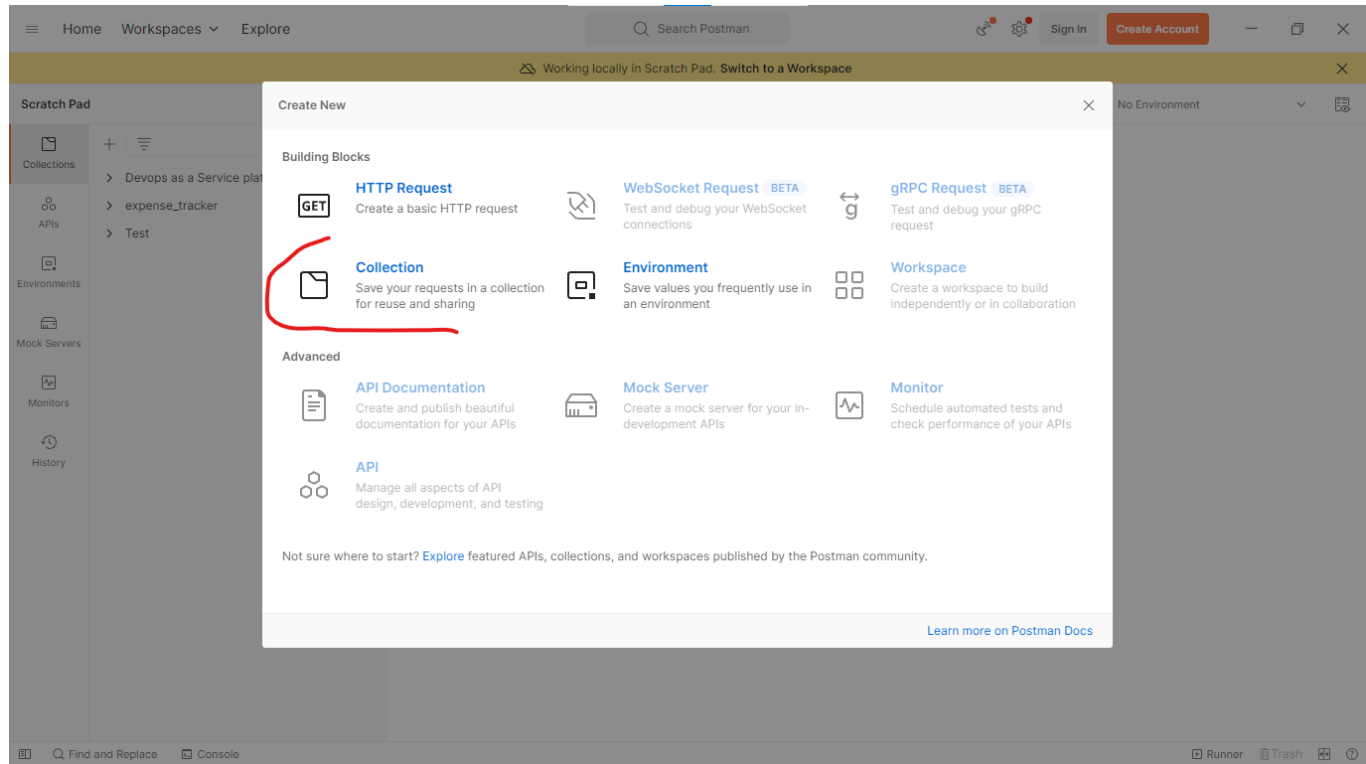
Once this is done, you're ready for the next step, which is testing the API with Postman.

## Testing on Postman

Open up the Postman application, Click on the "New" button on the left sidebar.



Select "Collection" from the dialog box that pops up



A collection in Postman is used to group all the endpoints that are tested for a particular site or application. You can create folders for different modules/resources of the application you're testing.

Clicking this will create a new collection which you can name, call it "Campay API Test".

Seeing as each of our requests will require authentication, it is a good idea to include this authentication at the level of the collection, that way all, the requests will inherit it.

## Creating Collection variables

We'll first start off by creating variables for our API\_URL and the Token.

Go to the variables tab of the "Campay API Test" collection.

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad

CamPay API Test

Variables

These variables are specific to this collection and its requests. [Learn more about collection variables.](#)

VARIABLE	INITIAL VALUE	CURRENT VALUE	...	Persist All	Reset All
<input checked="" type="checkbox"/>					
Add a new variable					

① Use variables to reuse values and protect sensitive data  
Store sensitive data in variable type secret to keep its values masked on the screen. [Learn more about variable type](#)  
Work with the current value of a variable to prevent sharing sensitive values with your team. [Learn more about variable values](#)

When there, you will fill in values for the API\_URL ( `https://demo.campay.net/api` ) and your token (the one you copied from the CamPay demo site). It should like something like below when done.

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad

CamPay API Test

Variables

These variables are specific to this collection and its requests. [Learn more about collection variables.](#)

VARIABLE	INITIAL VALUE	CURRENT VALUE	...	Persist All	Reset All
<input checked="" type="checkbox"/> API_URL	https://demo.campay.net/api	https://demo.campay.net/api			
<input checked="" type="checkbox"/> TOKEN	<paste-your-token-here>	<paste-your-token-here>			
Add a new variable					

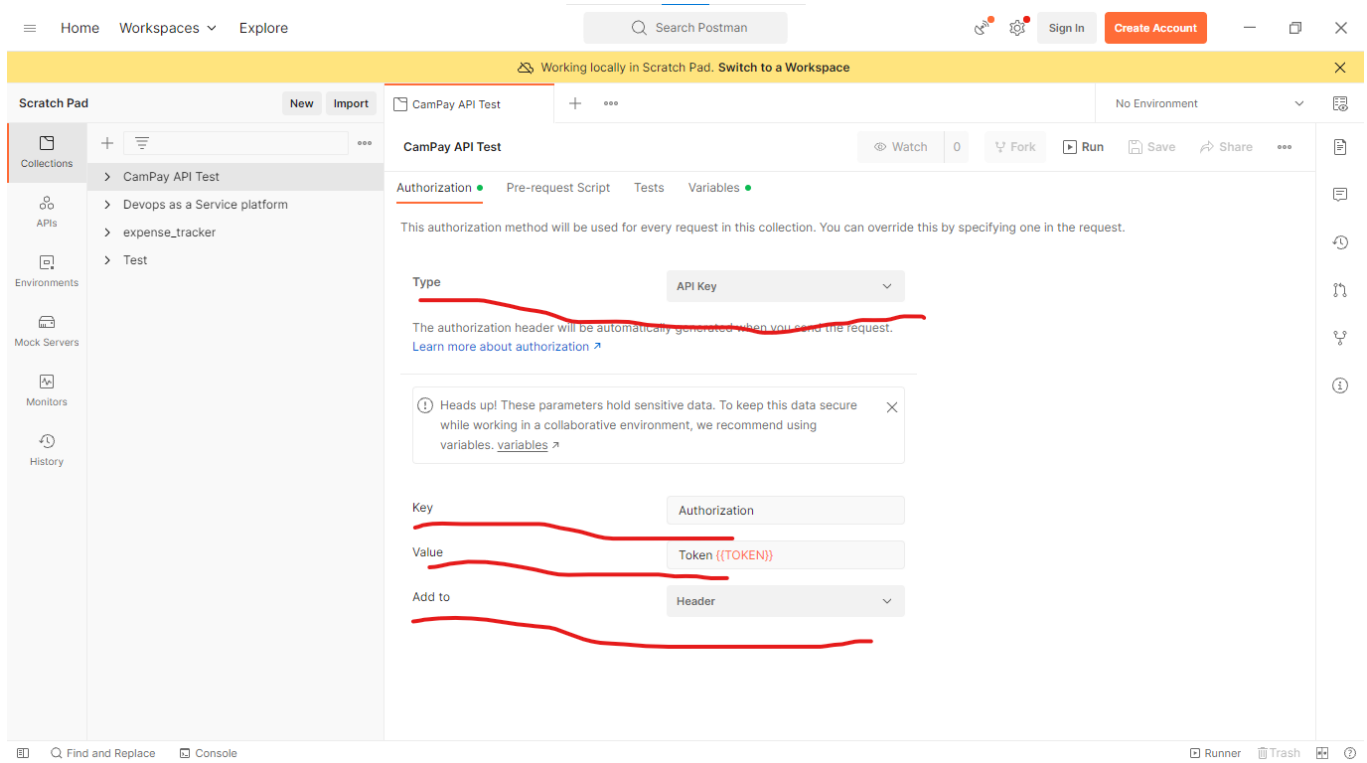
① Use variables to reuse values and protect sensitive data  
Store sensitive data in variable type secret to keep its values masked on the screen. [Learn more about variable type](#)  
Work with the current value of a variable to prevent sharing sensitive values with your team. [Learn more about variable values](#)

When done filling the values, save by pressing Ctrl+S.

# Configuring authentication

Once you've put in your token and api\_url variables, we have to tell Postman to include our auth details in the **request headers** of all of our requests.

Go to the Authorization tab of this collection. Put in the values indicated on the highlighted fields.



Save.

Once this is complete, you are done with the initial setup of the Collection. You can now start writing requests.

## Adding a request

We will start by adding a request for "Collecting Payment". You can read the documentation [here](#). This endpoint is used to request payment from users.

Right click on the collection and select the "Add request" option. This will create a new request.

You can rename the request by clicking on the pen icon next to the request name.

Go to the "Headers" tab of the new request and add the header shown in the image (Content-Type with value application/json).

Home Workspaces Explore Search Postman Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import POST Collect Payment

CamPay API Test / Collect Payment

POST {{API\_URL}}/collect/ Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Headers 9 hidden

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> Content-Type	application/json			

Response

Click Send to get a response

Once this is done, go to the body tab, select the raw option and put in JSON data as shown below.

Home Workspaces Explore Search Postman Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import POST Collect Payment

CamPay API Test / Collect Payment

POST {{API\_URL}}/collect/ Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

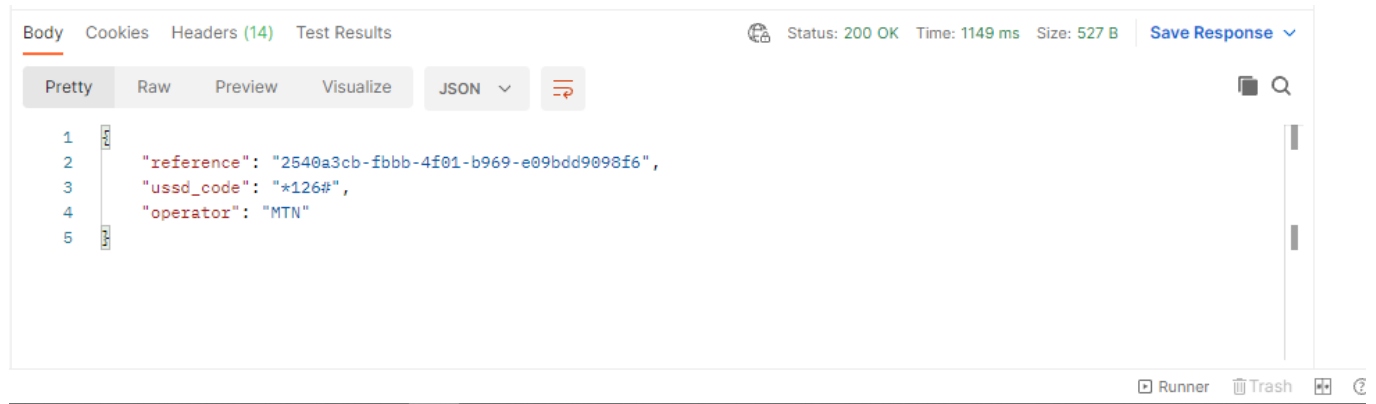
```
1 {
2   "amount": "50",
3   "currency": "XAF",
4   "from": "2376XXXXXXX",
5   "description": "Test",
6   "external_reference": "",
7   "external_user": ""
8 }
```

Response

Click Send to get a response

Don't forget to put a valid phone number in the "from" key.

Click the Send button and then save (Ctrl+S).



The response from this request looks something like above. Copy the reference code returned, you will need it for later.

Add another request for the transaction status. Use this endpoint to check for the status of an initiated transaction, use the reference you copied above.