POSTMAN

POSTMAN

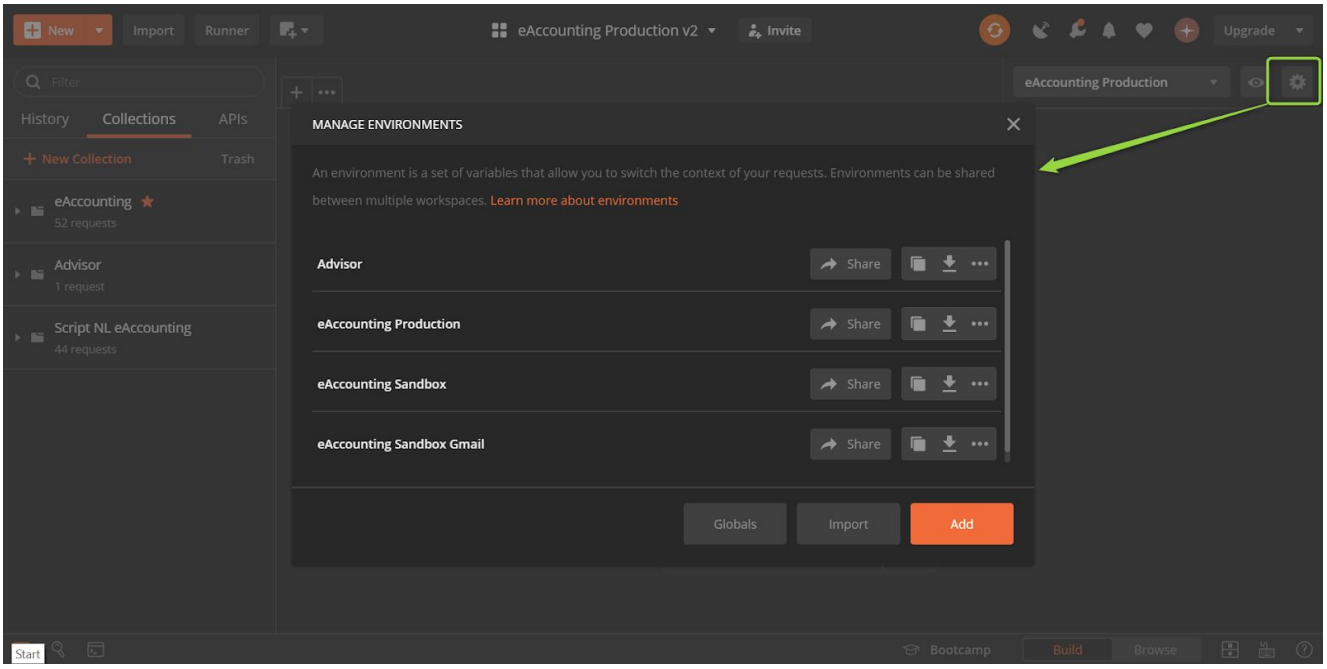
Postman is a tool for HTTP request creation and catching HTTP responses. It also enables automatic testing of APIs, supports Authentication protocols (like OAuth 2) and many more.
The tool is available in the browser or as a standalone application. In this document, the general use case of Postman in the context of Visma eAccounting API will be considered.

Environments

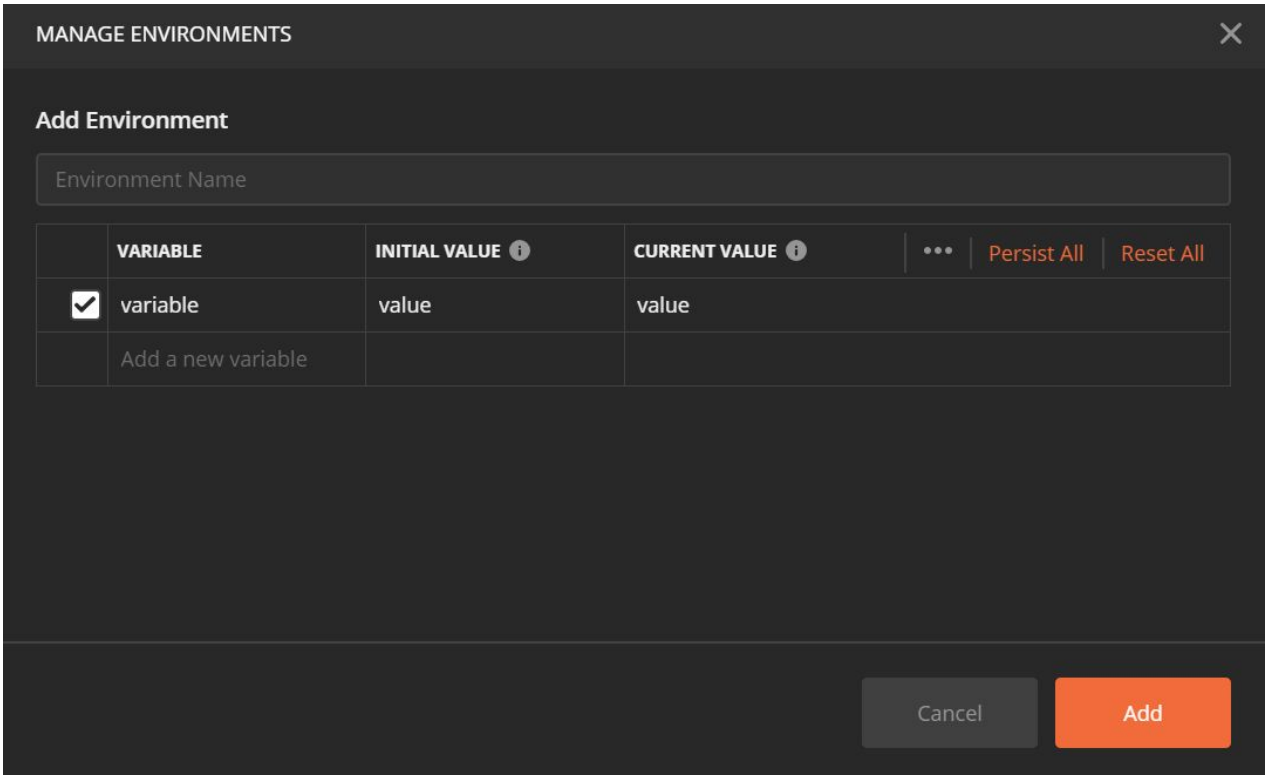
As it was written in the API **client configuration** document, there are two environments available in eAccounting API (Sandbox and Production).
Postman has the possibility to define different environments in its Environment module. Basically, in Postman understanding, the environment is a set of specific variables that could be called up when this environment is selected on the menu.

Creating an Environment

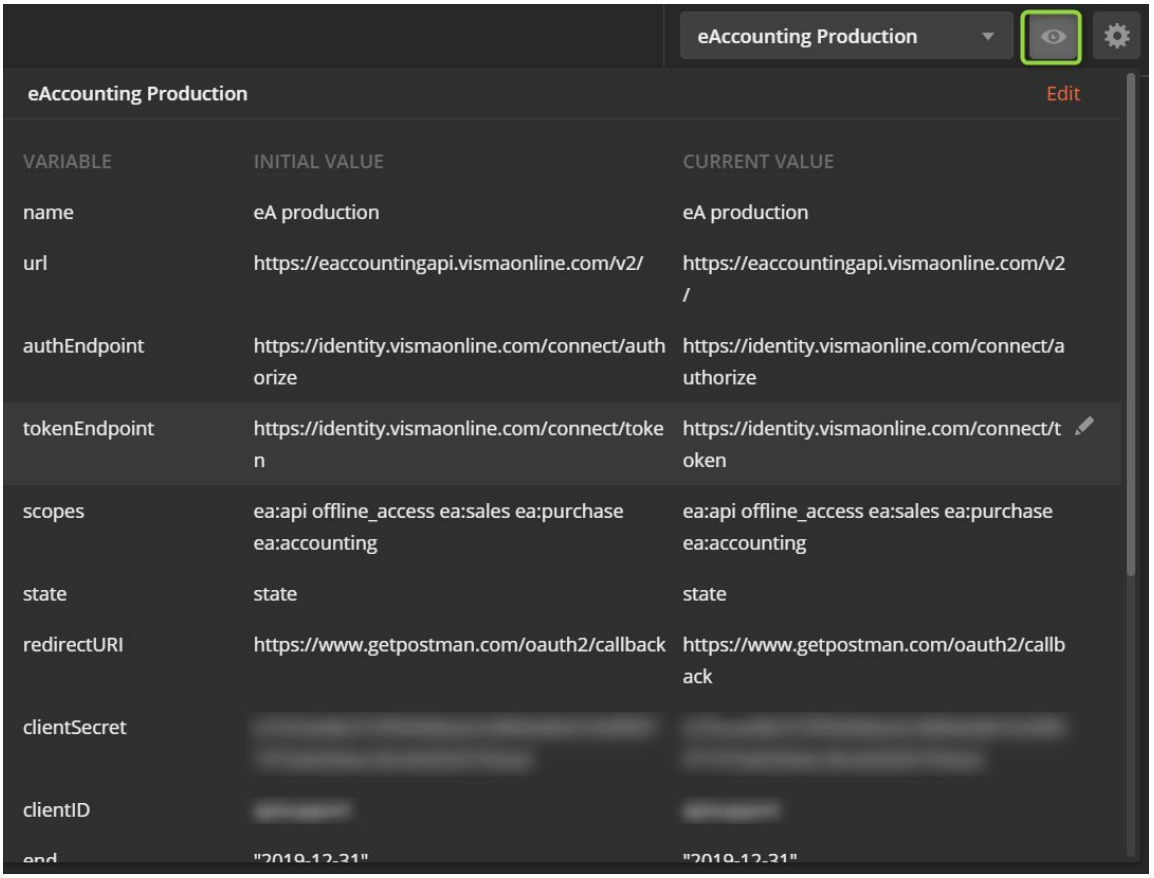
In order to create a new environment, click the gear wheel button in the top right corner of UI.



After clicking on the Add button, it is possible to define the new Environment with a set of variables, its initial and current value. Those variables will be accessible later.



After Environment creation, there is a possibility to see a preview of it, by clicking an Eye icon near the gearwheel button (top right corner).



Variables inside Environment - recommendation

This is our recommendation for a set of variables inside the environment, for working with eAccounting API:

Variable	**Notes:**
authEndpoint	URL for the authorization endpoint
tokenEndpoint	URL for the token endpoint
scopes	List of scopes wished to be used.
state	A unique string that is passed back upon completion.
redirectURI	In the case of postman, the following is recommended: https://www.getpostman.com/oauth2/callback
clientSecret	Received client secret
clientID	The ID of the client.

Authentication

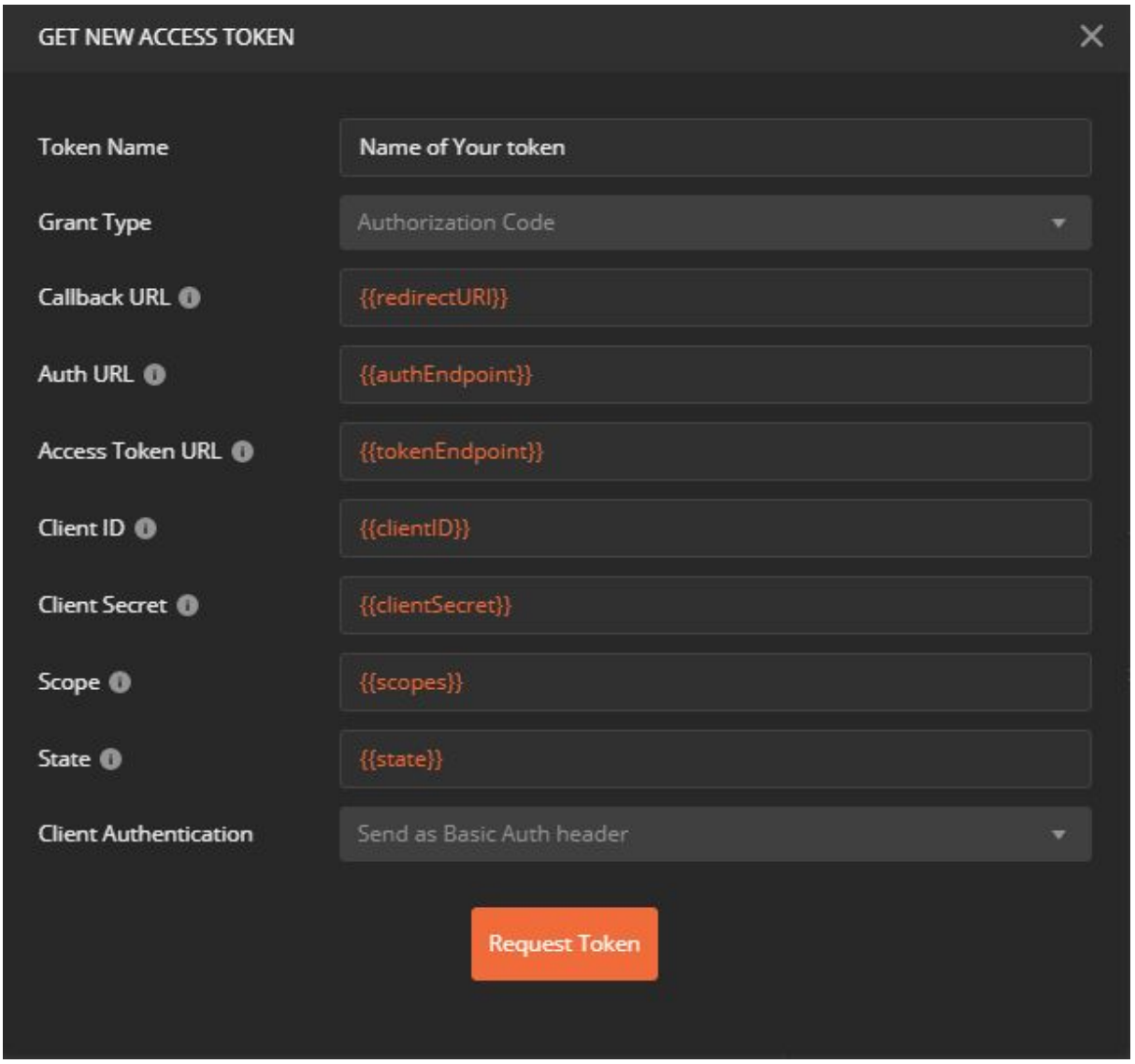
Not all APIs require authentication, but some do.
There are many frameworks an API server can use for authentication, and it could be customized in many ways. All necessary information can be found in the documentation of specific API.

Postman supports a variety of authentication frameworks. One of them is OAuth2.
Therefore, it is possible to do the authentication automatically with this tool.
In order to do that:

1. Create a new tab with a plus sign.
2. Click the Auth option.
3. Choose OAuth2.0 as a TYPE
4. Add auth data - define where auth data will be added in the request after the Authentication process is completed (in case of eAccounting API it should be Request Headers).
5. Click Get New Access Token.

Below, there is a picture with all the mentioned steps:

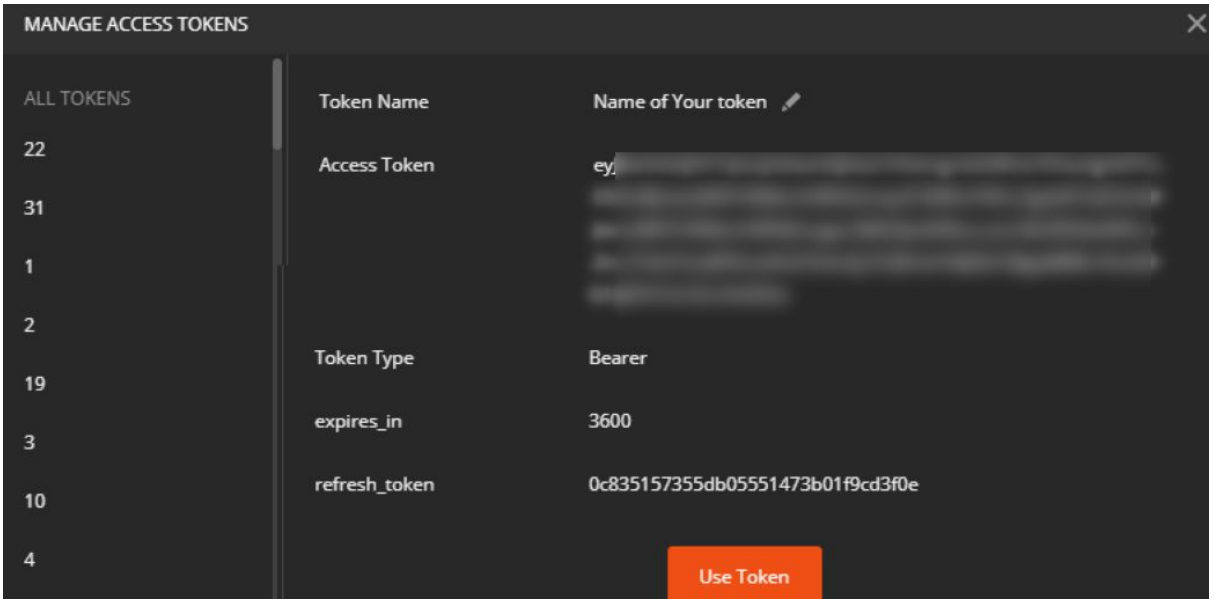
A window will be prompted where you should enter your parameters:



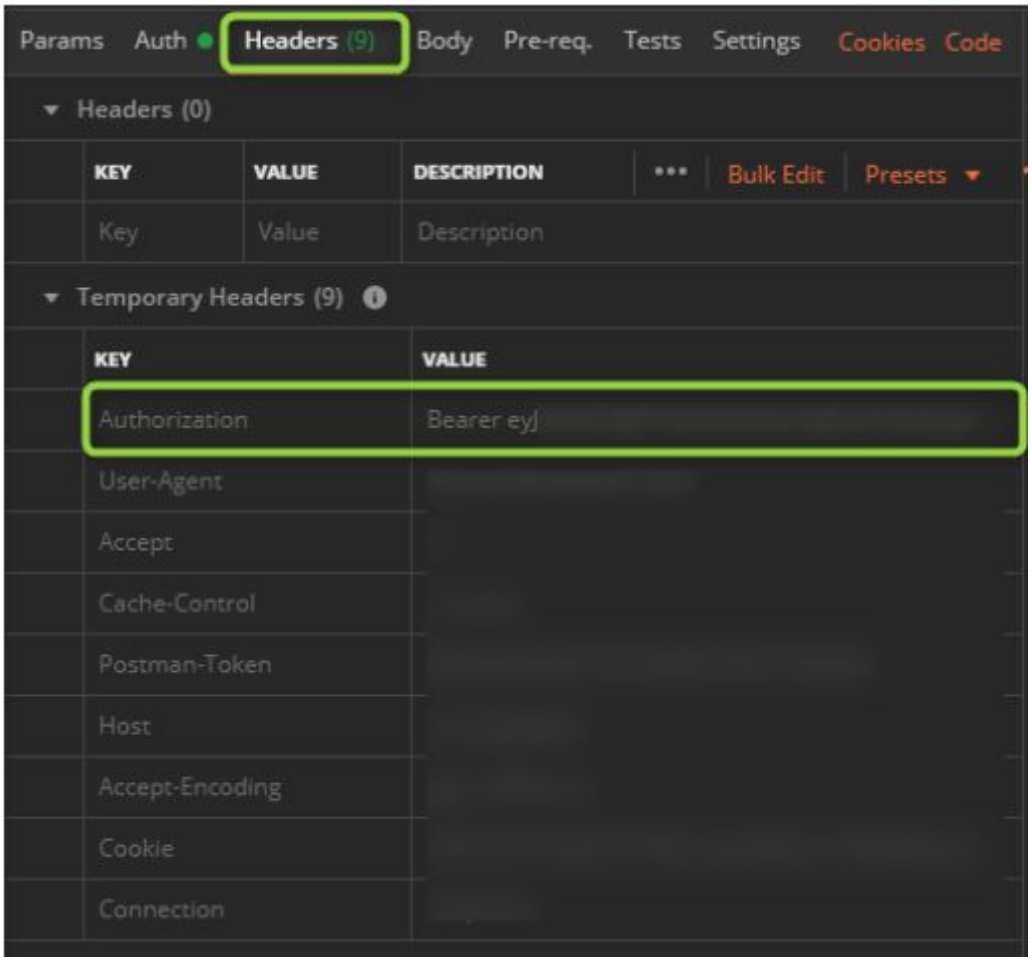
In this window the following information should be specified:

1. Token Name (freely to define).
2. Grant Type (Authorization Code for eAccounting API)
3. Rest of parameters obligatory from OAuth2 framework perspective. It is possible to use variables defined in the previous chapter. In order to use them, they should be inserted in the double curly brackets: **{{variable_name}}**.
4. Client Authentication - defines a way how client credentials (client_id and client secret shall be send)

After pressing the Request Token button, OAuth2 flow will be started. As a result of that flow, Access Token should be received and displayed for the user.



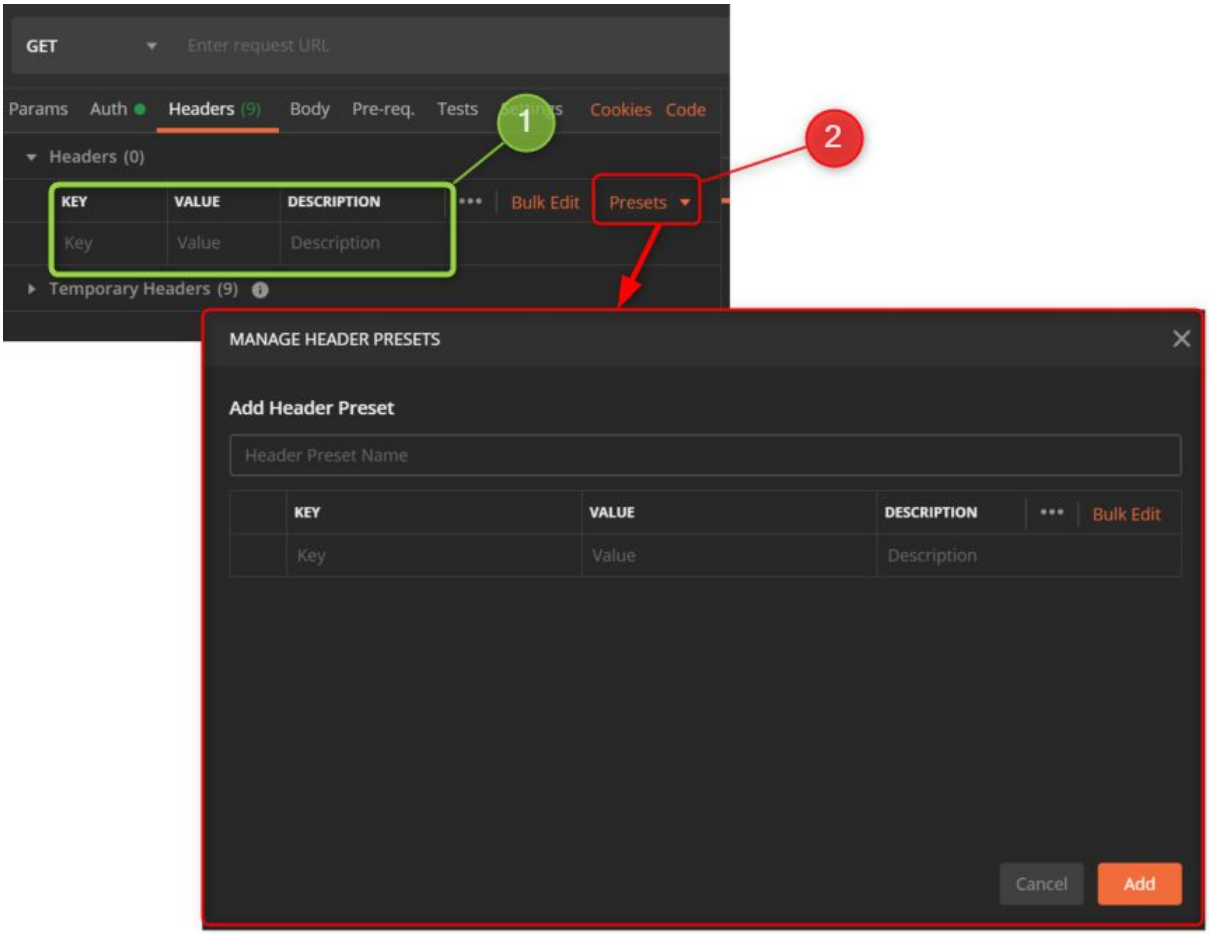
After pressing use token, Postman will automatically add an access token in the Header of the request. Token can be found in the Headers section of the request (after request is executed) as the temporary header (which is not saved in the request later).



Requests

Headers

It is also possible to define static headers of the HTTP Request. It can be done either manually(1) or template with headers with Presets option(2) can be created:



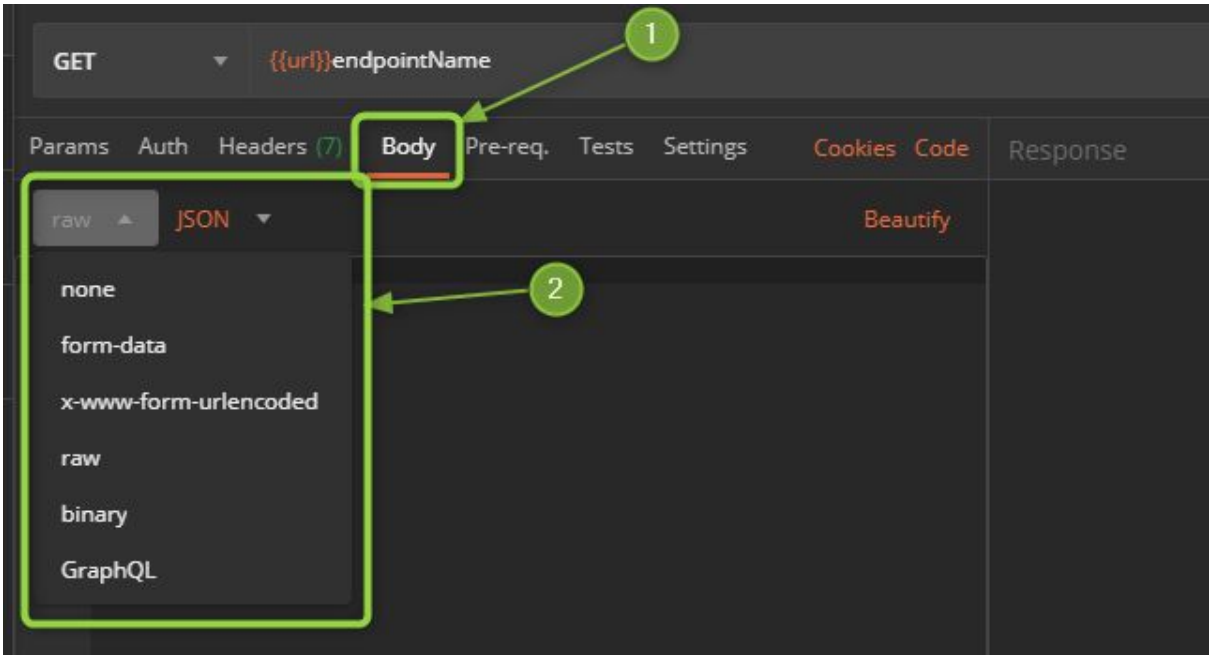
With the preset option, it is possible to define a set of headers which can be easily assigned later on in each future request (i.e. like an access token with Manage tokens option).

Request URL

In the request URL, the requested endpoint can be defined. Within this field there is also a possibility to use variables defined in the Authentication part. Syntax is the same (double curly brackets):

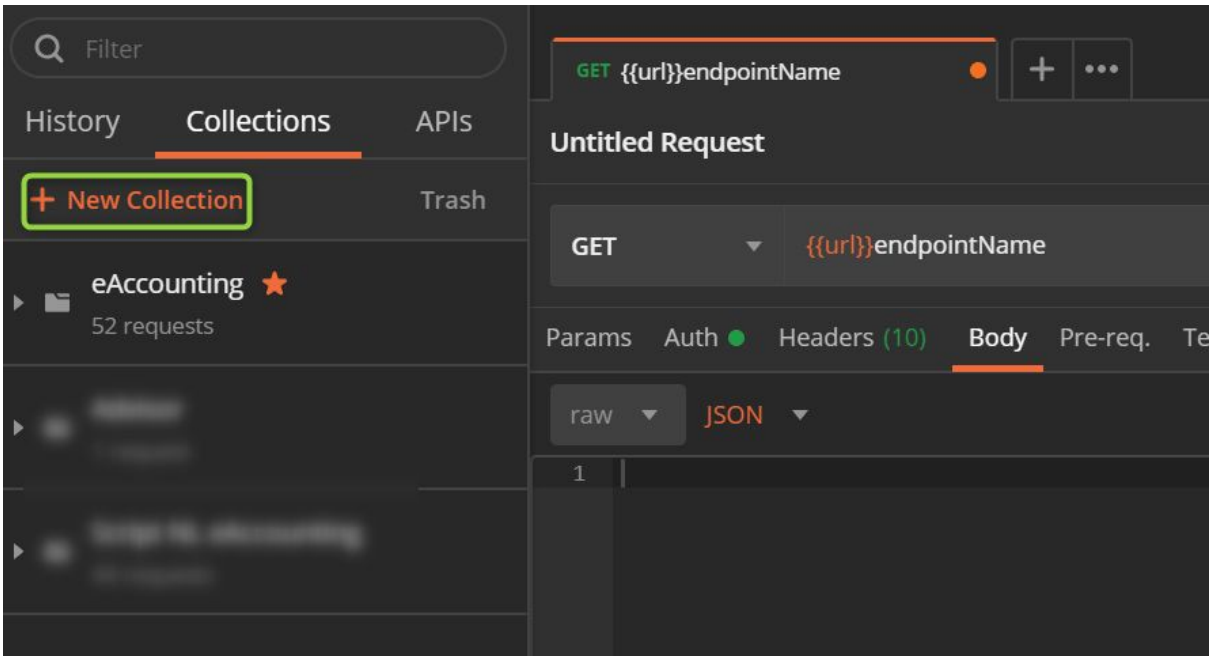
Request Body

Body of the Request is usually obligatory, when there is a need to add something in the server, so in case of POST or PUT request. Body can be added after pressing a Body field in the Postman (1). It is also possible to specify the content of the body(2). Parameters set here, will be automatically added as the temporary header of the request.

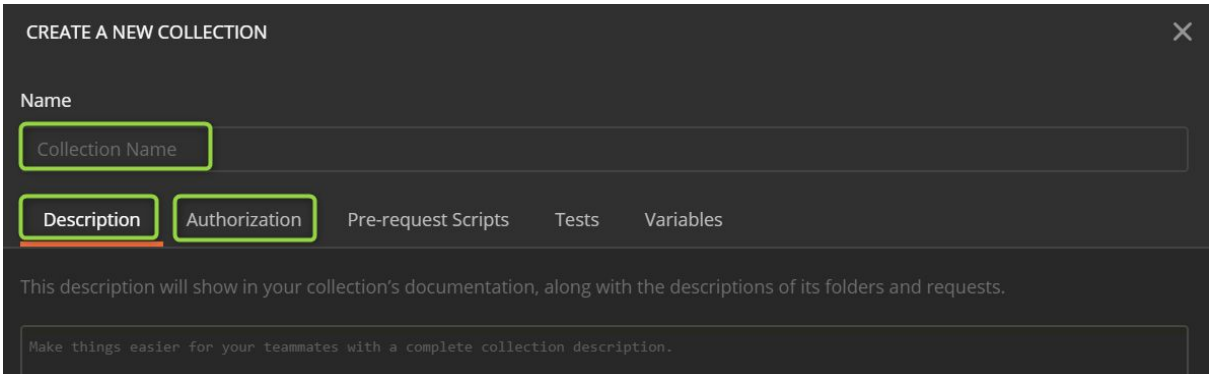


Collections

Collections is a feature, which allows storing and managing groups of requests in one place (folder). New Collection can be created with the help of New Collection button:

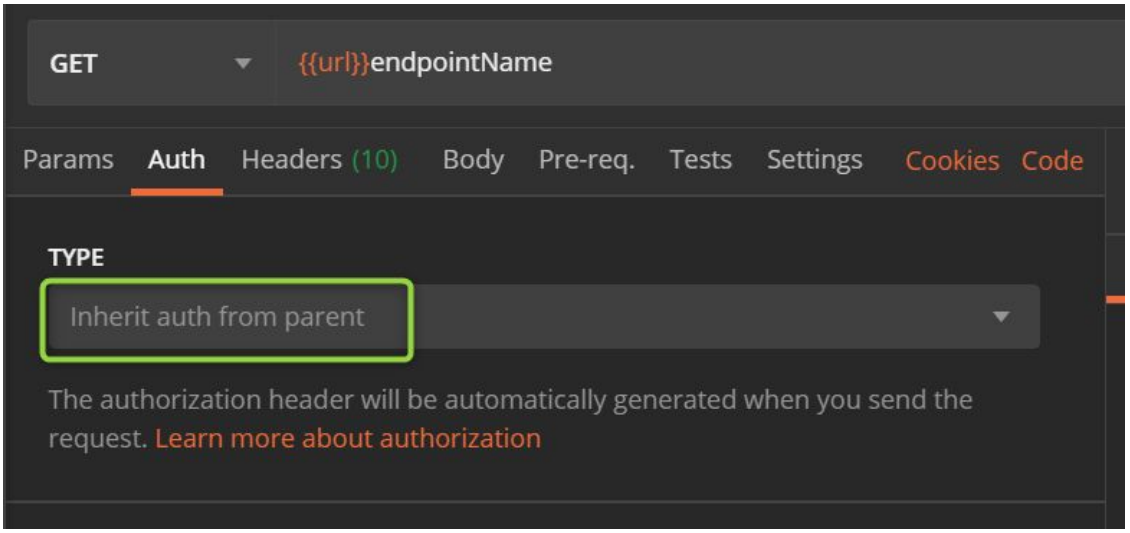


Configuration window will be shown, in which there is possible to put a name of the Collection, Description and Authorization settings for the whole collection.



Settings for the authorization are the same as in the Authentication section, but if created inside the collection settings may apply to all the collection members.

In order to configure this functionality, in each request which belongs to such a collection, in Auth section Inherit from parents option has to be chosen:



Endpoints

In this section examples of eAccounting endpoints requests are shown. All endpoints are available to explore in our Swagger:
For Production environment:
https://eaccountingapi.vismaonline.com/swagger/ui/index#/
For Sandbox environment:
https://eaccountingapi-sandbox.test.vismaonline.com/swagger/ui/index#/

When You are creating a request against eAccounting API, there is necessary to put an access token in the header. If You obtain an access token, Postman will automatically add this token as the temporary Header as it was written in the previous chapter.

It is possible to add other Headers, but they are optional for eAccounting API:
Those headers are:

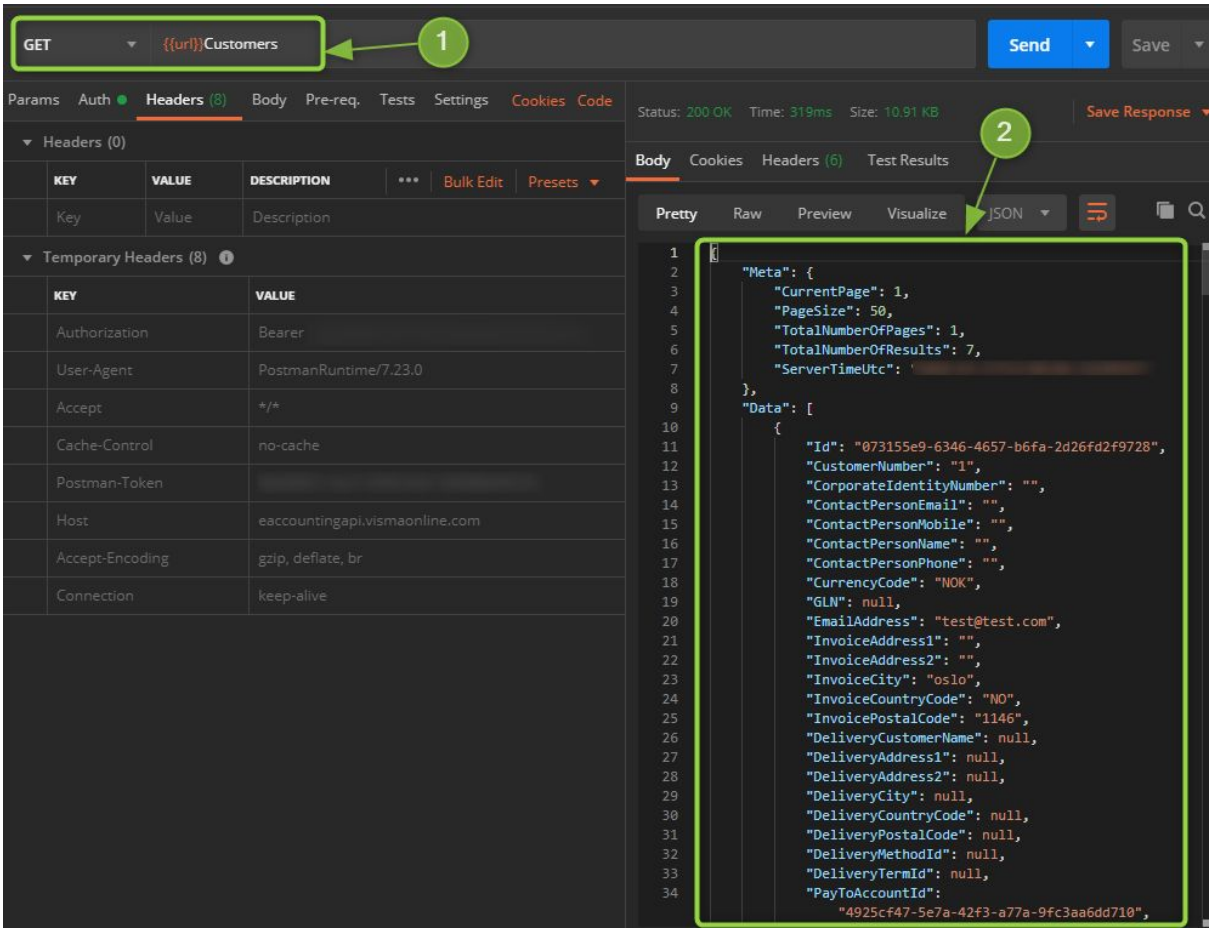
- ➔ Accept: application/json
- ➔ Content-Type: application/x-www-form-urlencoded

Examples

Below, examples of requests (GET, POST):

GET https://eaccountingapi.vismaonline.com/v2/customers

In the Postman, POST method is chosen, as the HTTP head endpoint Customers, together with URL variable (1) and HTTP Response Body from the server (2).



POST
https://eaccountingapi.vismaonline.com/v2/customers
{
"CustomerNumber": "112",
"EmailAddress": "john@visma.com",
"InvoiceAddress1": "Vangjordet 44",
"InvoiceCity": "Vestby",
"InvoiceCountryCode": "NO",
"InvoicePostalCode": "1540",
"Name": "Lars",
"MobilePhone": "98639741",
"TermsOfPaymentId": "0574e7db-1e23-41ed-bf80-647e9830e458",
"VatNumber": "877654321",
"IsPrivatePerson": false,
"IsActive": true,
"ForceBookkeepVat": true,
"SalesDocumentLanguage": "NO"
}

In the Postman, POST method is chosen, as the HTTP head endpoint Customers, together with URL variable (1), HTTP Request Body (2) and HTTP Response Body from the server (3).