User Guide   
SAP API Management dev portal

Table of Contents

[1. SAP API dev portal 3](#_Toc64048915)

[1.1 Self Registration 3](#_Toc64048916)

[1.2 API Catalog 3](#_Toc64048917)

[1.2.1 API References 4](#_Toc64048918)

[1.2.2 Details 6](#_Toc64048919)

[1.3 Testing API from dev portal 7](#_Toc64048920)

[2. Test an API from Postman. 9](#_Toc64048921)

[2.1 Basic Authorization 9](#_Toc64048922)

[2.2 API Proxy URL and API Key 9](#_Toc64048923)

# SAP API dev portal

## 1.1 Self Registration

As a user you can self-register on the SAP API Management’s Dev portal (here by referring as dev portal) using the following link:

https://q81.apibhubenterprise.cfapps.eu10.hana.ondemand.com[/?saml2idp=accounts.sap.com](https://q81.apibhubenterprise.cfapps.eu10.hana.ondemand.com/?saml2idp=accounts.sap.com)

It is the dev portal on quality environment of SAP API Management. Use your danone’s email ID to register. Your registration request is sent to the GTI – SAP DevOps team who will approve the request. If your request is not approved for a long time, send an email to [sap-cloud-integration@danone.com](mailto:sap-cloud-integration@danone.com)

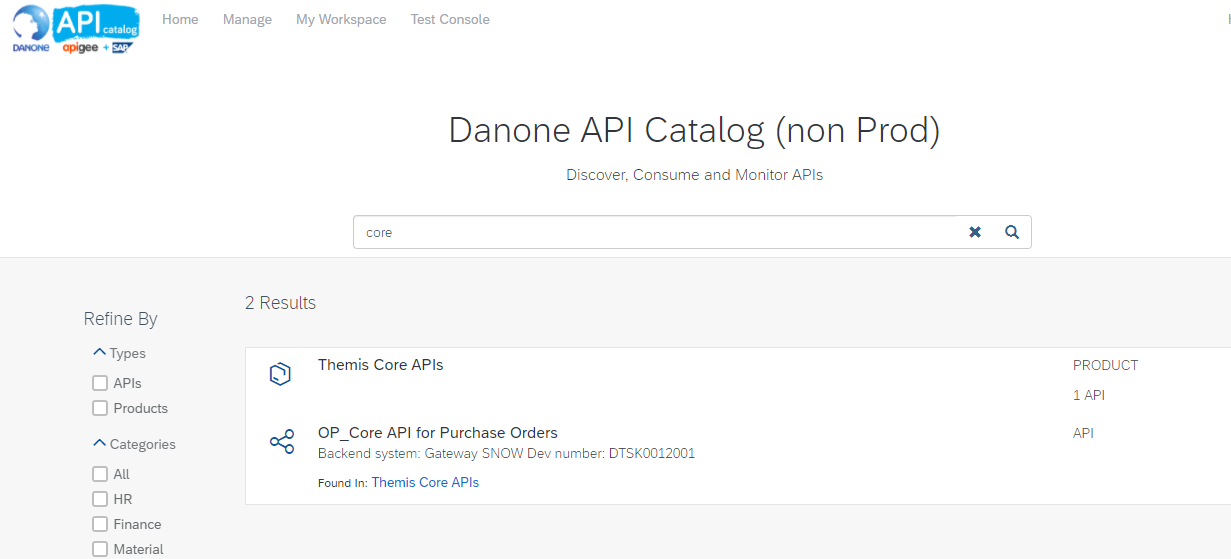
## 1.2 API Catalog

You will receive a confirmation email after your self-registration request is approved.

After logging into the dev portal, you will see the home page by default. Home page displays the API Catalog that is available. It contains all the APIs available and their products in which they belong. You can either search for the API by using keywords or select categories on the left to navigate to the desired API.

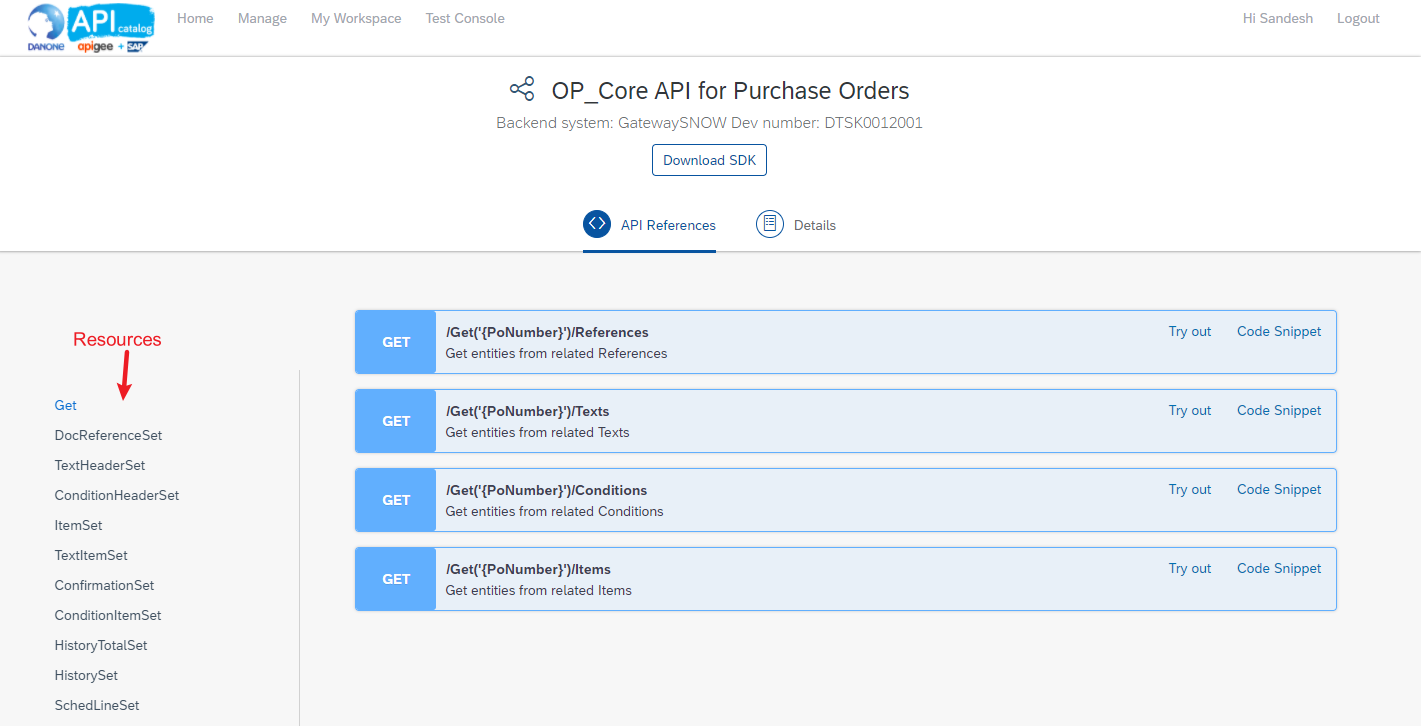
Concepts of Product and API:

* Product – It is a package or a collection of APIs. For example, all the Themis Core APIs can be found under the product: Themis Core APIs
* API – As the name itself suggests, it is the artifact which contains the documentation of the API, the end-point which the consumer application should use



In the search result in the above screenshot, you see there are two entities. One is product and next the API itself

Click on the API to see its documentation:

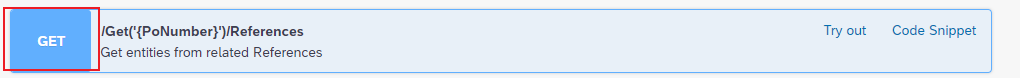


On the top you will see two tabs. API References and Details.

### 1.2.1 API References

This is based on the OpenAPI specification. Under this tab, you will see different resources available as part of this API. This information(metadata) generally comes from the backend service. In this example above, this information comes from the backend API that is developed on SAP Gateway.

For every resource there will be multiple API paths that are possible to be used as part of your API call. In the example above, **/Get(‘{PoNumber}’)/References** is a path. {PoNumber} should be replaced by a valid Purchase Order in your backend system.

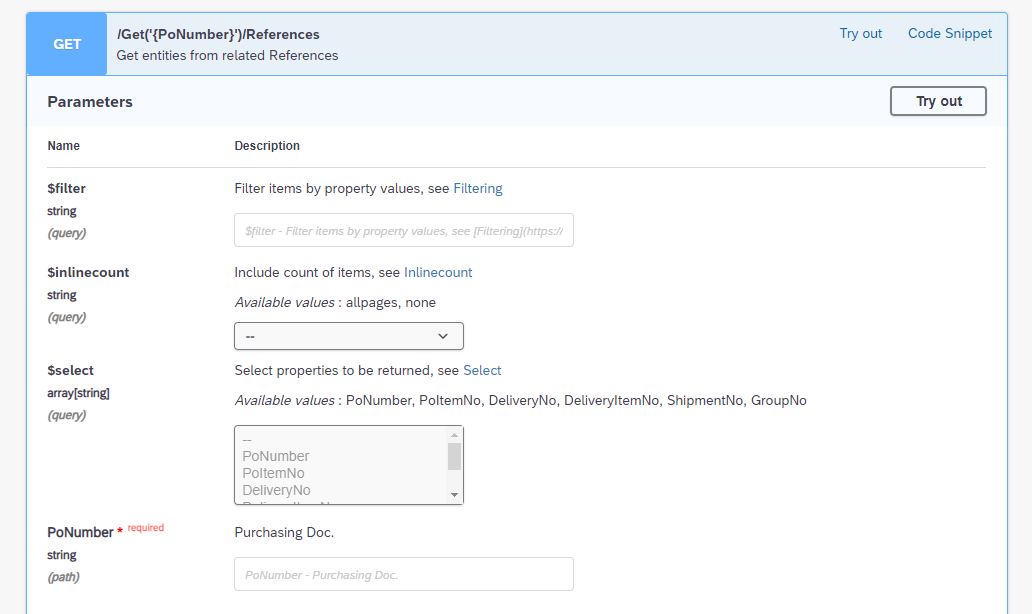


The Blue box in the above screenshot says about the HTTP method that is supported by this API resource. Here GET means, you can select or read data using this API path. Similarly, you may also see the following methods

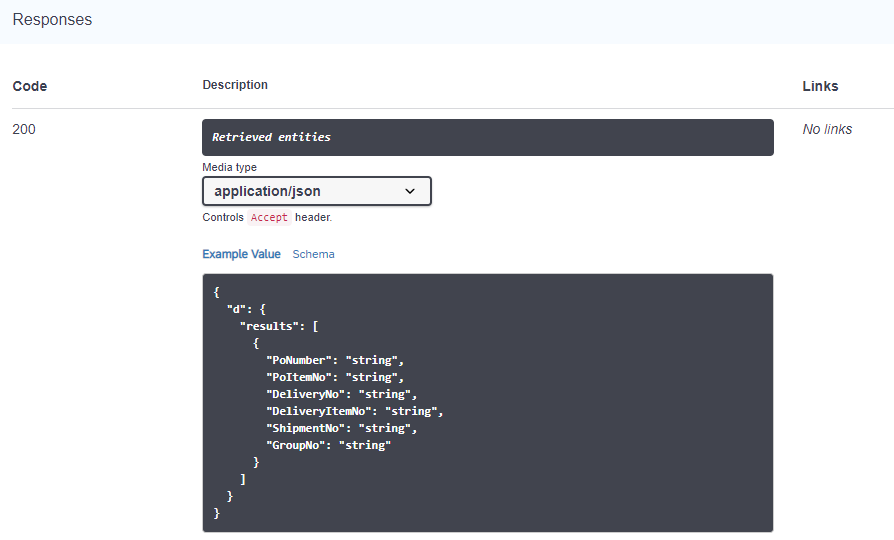
POST – You can create data in your backend system using this method  
DELETE – You can delete data in your backend system using this method

Such methods will only appear if the API supports them.

Click on the API Path to understand its definition and the fields you can expect in the request and response.

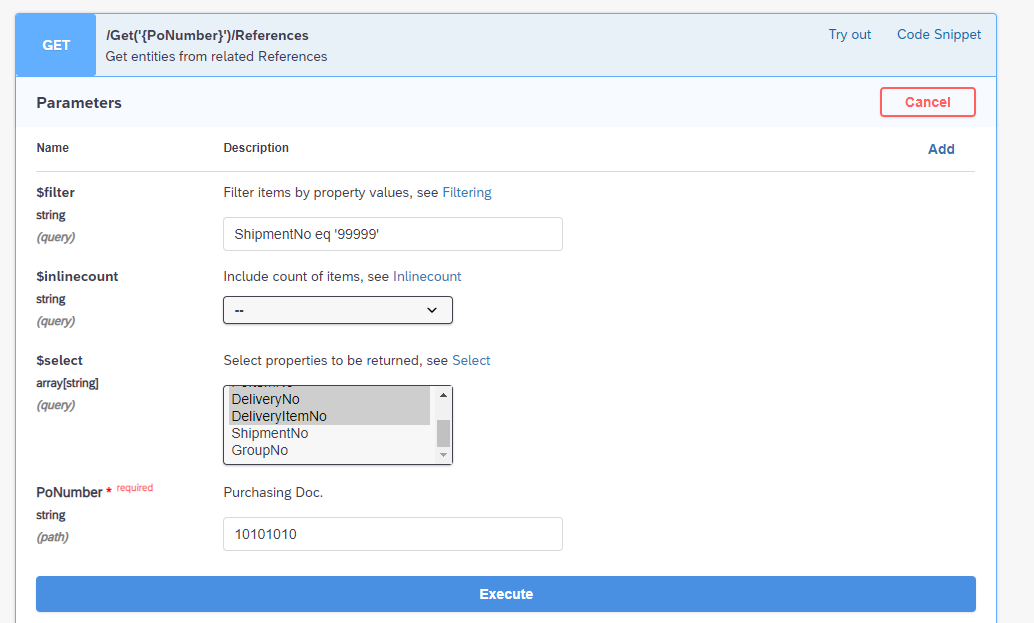


These are the different parameters that are supported by the API. As in this example, it is an OData API the parameters like $filter, $select are supported by standard. Please see documentation of OData to understand how they work -> [OData Documentation](https://help.sap.com/doc/5890d27be418427993fafa6722cdc03b/Cloud/en-US/OdataV2.pdf)

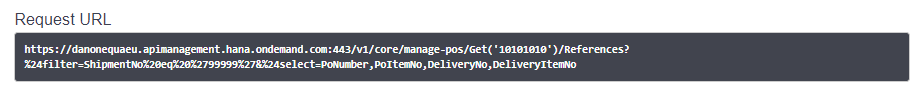


As seen in the above screenshot, you can see the fields that you can expect in the output and also their datatype.

Click on ‘Try Out’ to generate an end-point with which you can test the API later on. After clicking ‘Try Out’ fill the information of the parameters and click ‘Execute’ button. See below:

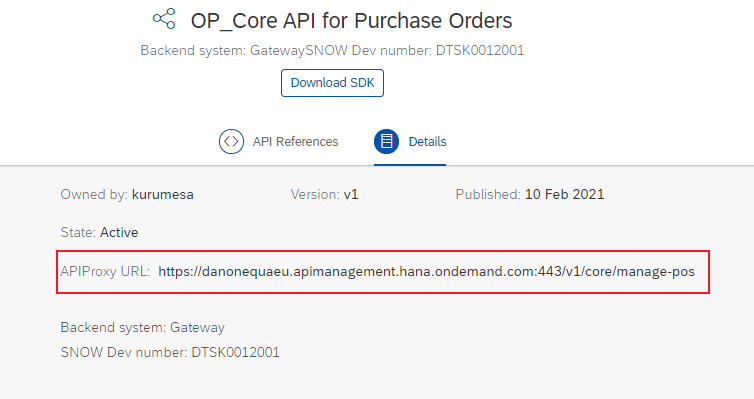


Clicking ‘Execute’ button will generate a Request URL which you can use it to test the API either from the test console inside the dev portal itself or from an external tool like Postman which is covered in the later part of this guide.



### 1.2.2 Details

Under the Details tab, you can see the base end-point URL of this API



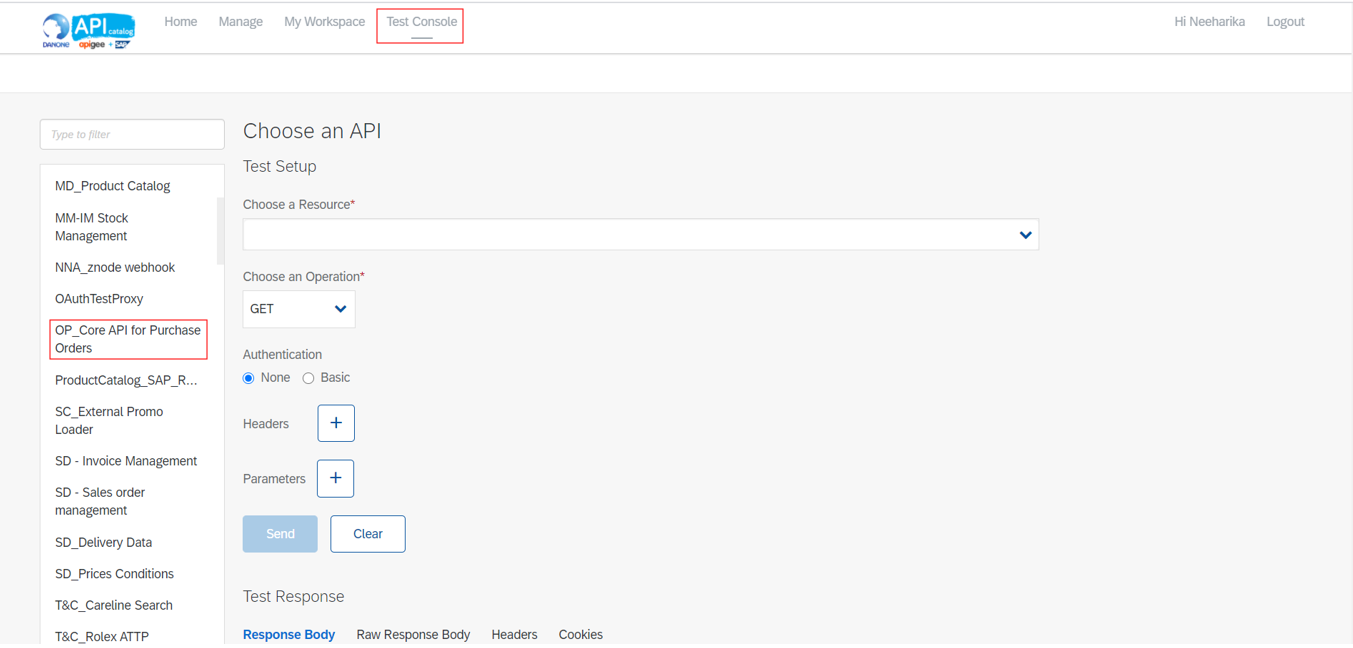
If it is a OData API, A technical developer who will consume this API on client side can understand the complete possibilities that this API offers by just having this base URL. He/She will query the metadata of this API by appending **/$metadata** to the base URL.

# 1.3 Testing API from dev portal

You will need an API Key before you can test. An API Key which works for a certain API need not necessarily work for another API. Get the API Key from DevOps team by sending an email to [sap-cloud-integration@danone.com](mailto:sap-cloud-integration@danone.com)

P.S: We are working on an approach to store all the API keys in a central repository which can be used to get the API key

* Navigate to ‘Test Console’ tab which is available on top of the page.
* Choose the appropriate API you would like to test. In this example, ‘OP\_Core API for Purchase Orders’ API is selected



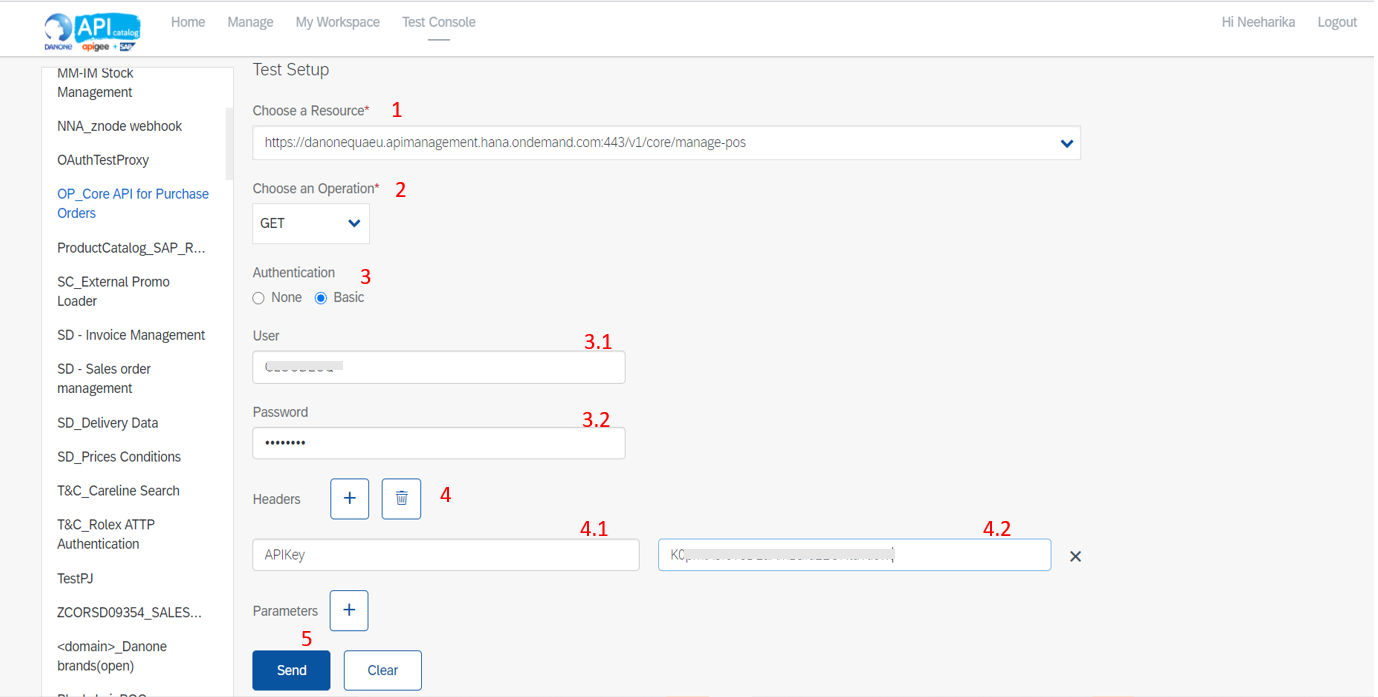
1. After choosing the API Proxy, in test setup space choose a resource from the drop list that you would wish to test for. These are the API paths discussed in the section 1.2
2. Choose an Operation, these operations are nothing but the HTTP methods therefore, choose an operation through which you can make a request to the backend system for the resource.
3. Authentication, if the backend system is secured with basic authentication check the radio button to ‘Basic’. You can get the Basic Authentication credentials from the security team.
   1. Fill the username.
   2. Fill the password.
4. Add header by clicking + beside ‘Headers’. Here you need to enter the api key for the header ‘apikey’ which you receive from the DevOps team.

4.1 Enter ‘apikey’

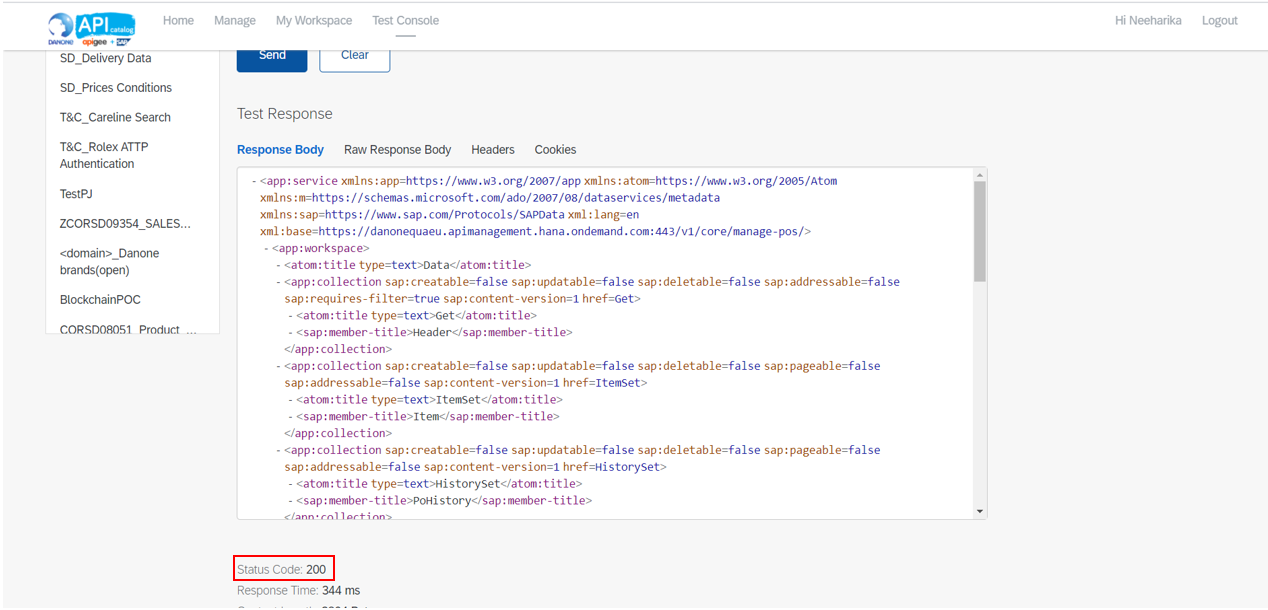
4.2 Enter the key

1. Now click on Send, to make a HTTP request for the resource available on the backend system.

TIP: If it is an OData API, you can get the data in json format for easy readability by setting a parameter ($format = json)



Once, you make a request to the backend system you receive the response body, and the status code 200 indicates that the request is succeeded.



Note:

Status codes:

2XX: indicates successful responses.

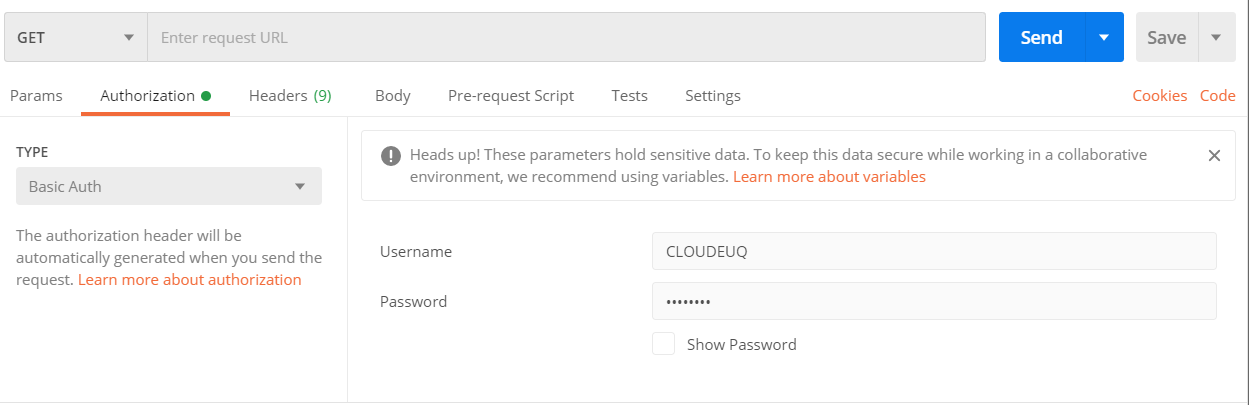
4XX: indicates client errors.

5XX: indicates server errors.

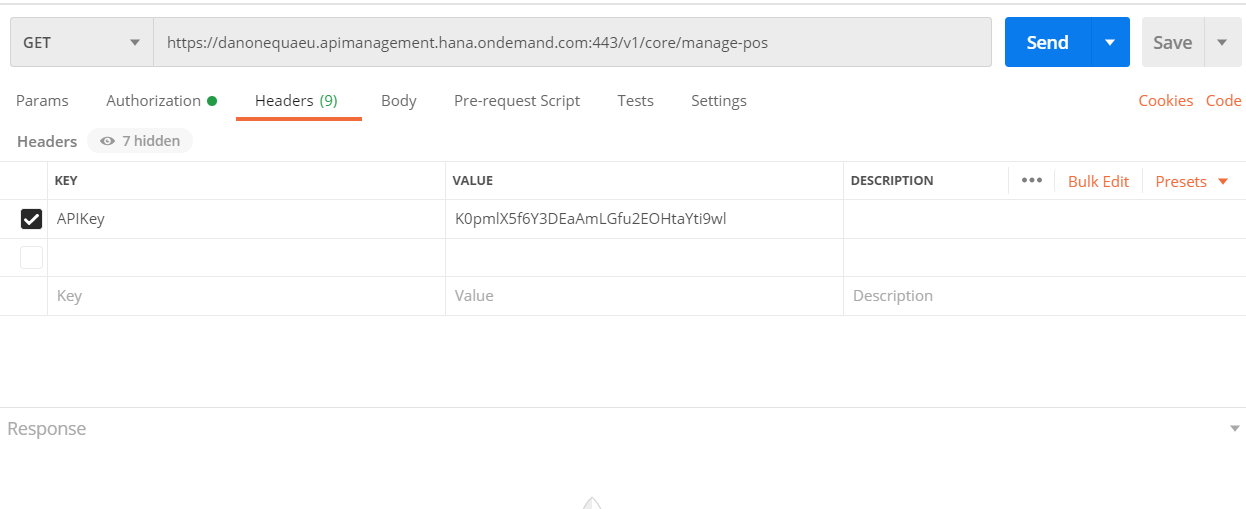
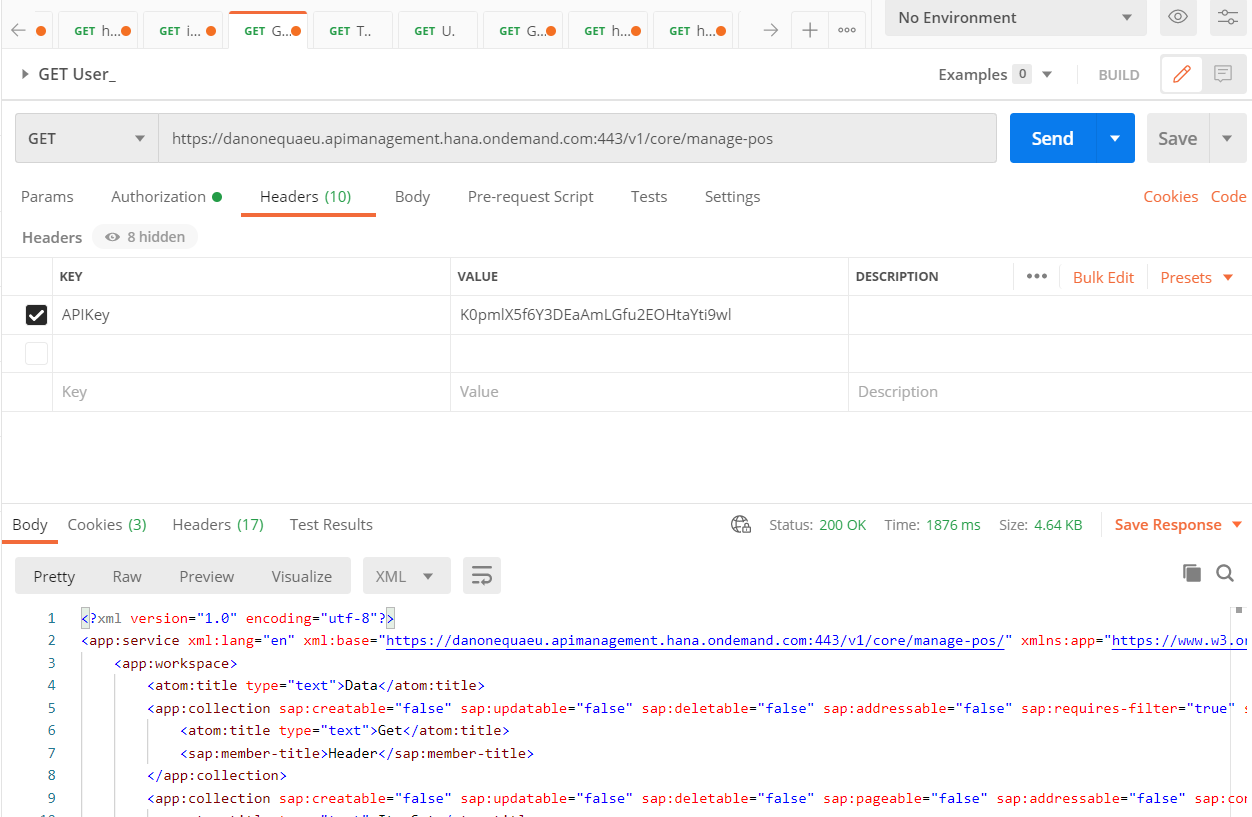
# Test an API from Postman.

Download and install postman tool from: <https://www.postman.com/downloads/>

## 2.1 Basic Authorization

* In ‘Authorization’ section choose authorization type as ‘Basic Auth’ and add the respective basic authentication credentials which are received from the security team.

## 2.2 API Proxy URL and API Key

* Fetch the API Proxy URL and API Key from the API Owner or from the API dev portal as described in section 1.
* Paste the API Proxy URL and choose the appropriate HTTP method to make a request to the backend system.
* In ‘Headers’ section add your API key parameter (APIKey) and its respective value, and click on ‘Send’.
* In Response Body you can find the response. If the status code is 2XX then your request is succeeded.