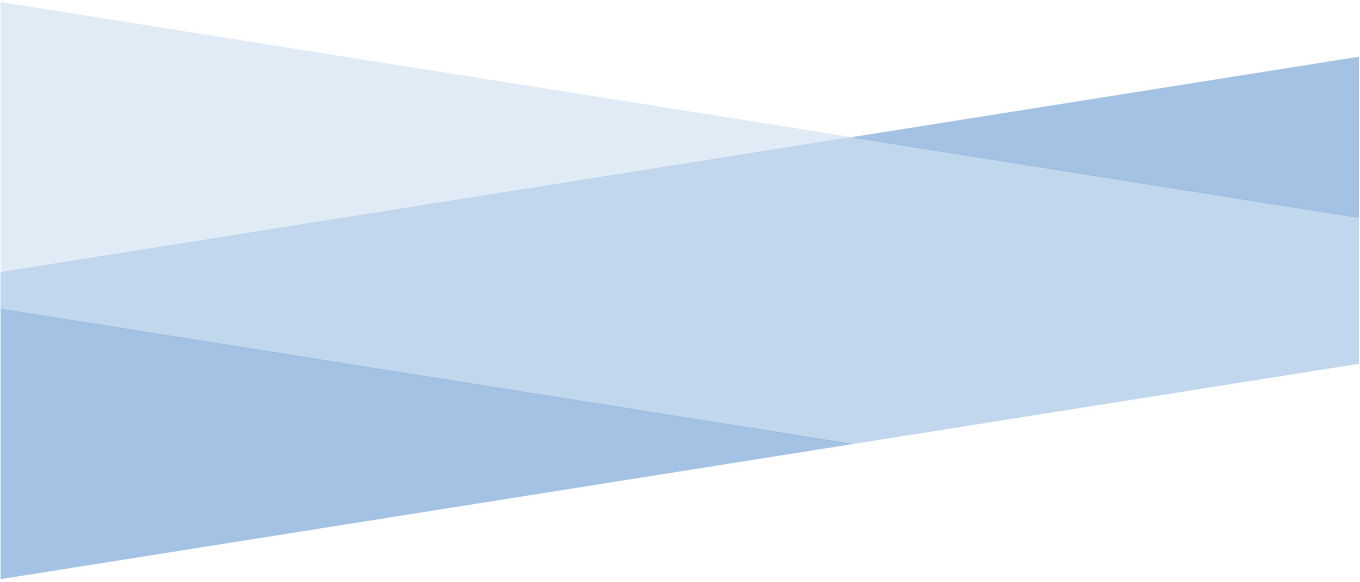
Reporting APIs Installation Guide



Apr 12, 2024

Installation Guide



Contents

[1. Technical Prerequisites 3](#_Toc164814682)

[1.1. **eFront Cube** 3](#_Toc164814683)

[1.2. **Reporting API plugin** 3](#_Toc164814684)

[2. Testing the installation 4](#_Toc164814685)

[3. How to test and validate 5](#_Toc164814686)

[3.1. Postman 5](#_Toc164814687)

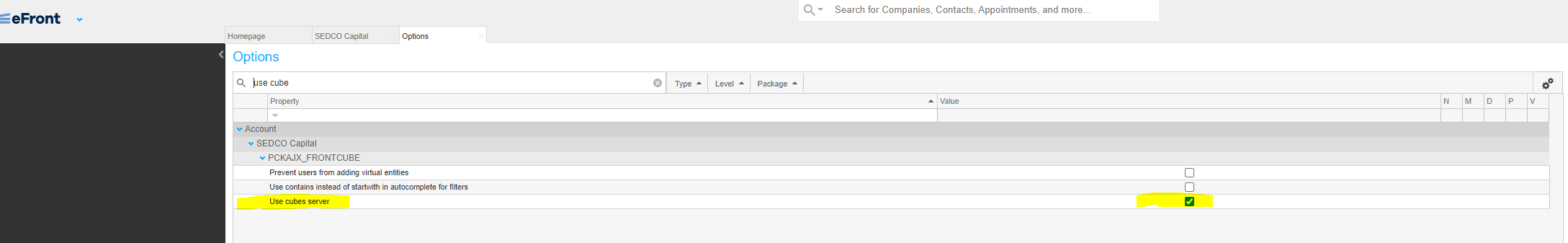
[3.2. Python 8](#_Toc164814688)

# Technical Prerequisites

## **eFront Cube**

Ensure eFront Cube package is activated in INPROC mode if not already activated with dedicated URL:

1. Add package PCKAJX\_FRONTCUBE.    
   (Menu >> Administration >> Account)
2. Tick the package option "Use cubes server"  
   (Menu >> Administration >> Options >> Search “cube”)
3. Do **NOT** modify anything in FrontAdmin



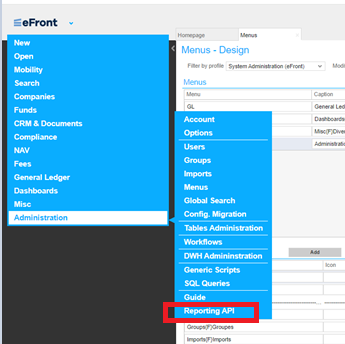
## **Reporting API plugin**

We need four things below followed by a restart.

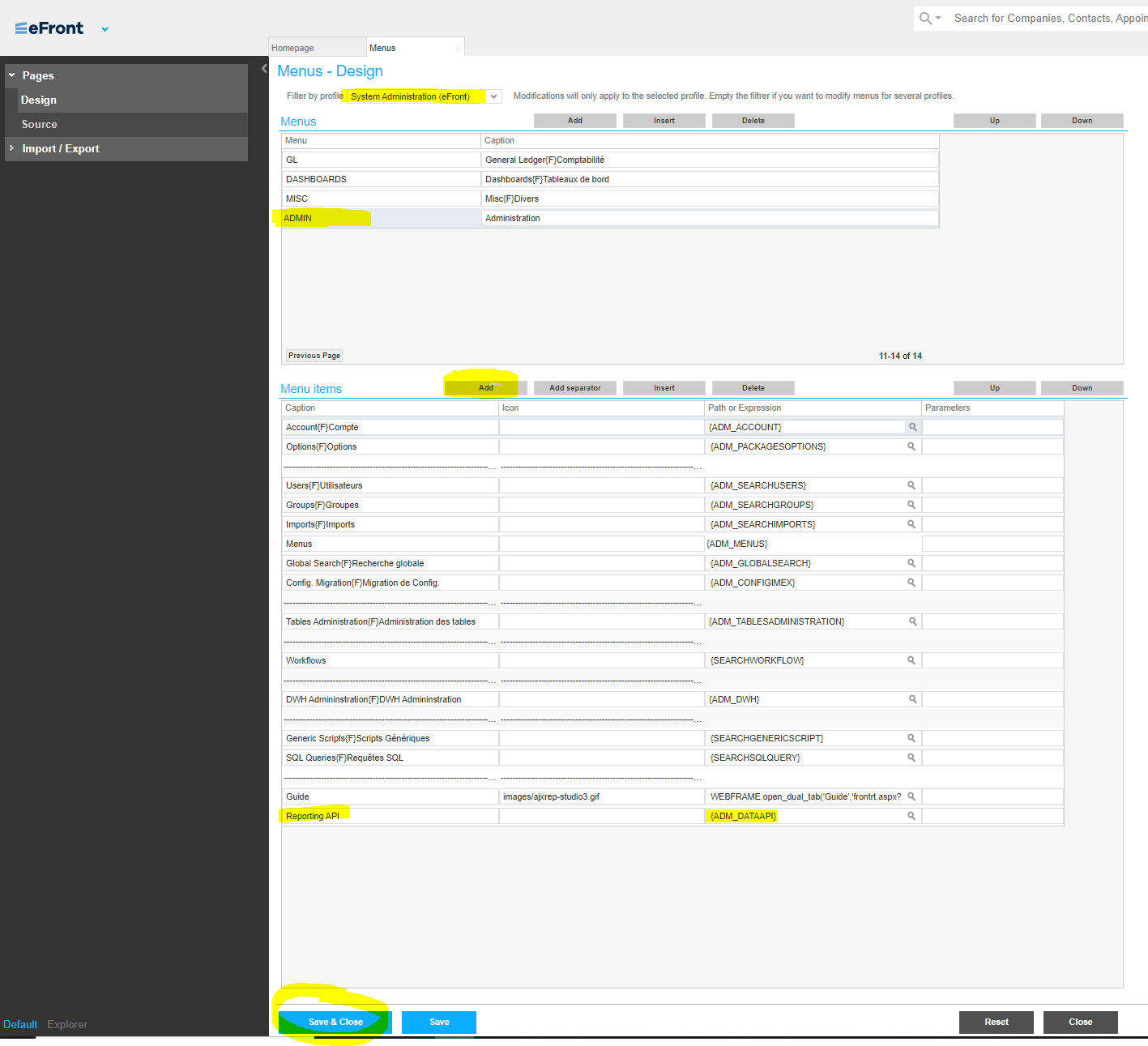
1. in the application, add package **PCKREPORTINGAPI**
2. on the server, create folder **website\plugins** and extract the contents of the binary zip into this folder.   
   (as of 28th Feb 2024, the last version is HydraPluginsPackage-2.28.0.5.zip)
3. Add "plugins" in web.config probing private paths (same as "cod"):   
   **<probing privatePath="cod;plugins;" />**
4. Accessing APIs via the browser via Windows Authentication is now disabled by default. To enable it, add the below setting to the web.config within <appSetings> part:   
   <add key="Hydra:AllowLegacyAspNetSessionAuthentication" value="true" />
5. Restart the website.

# Testing the installation

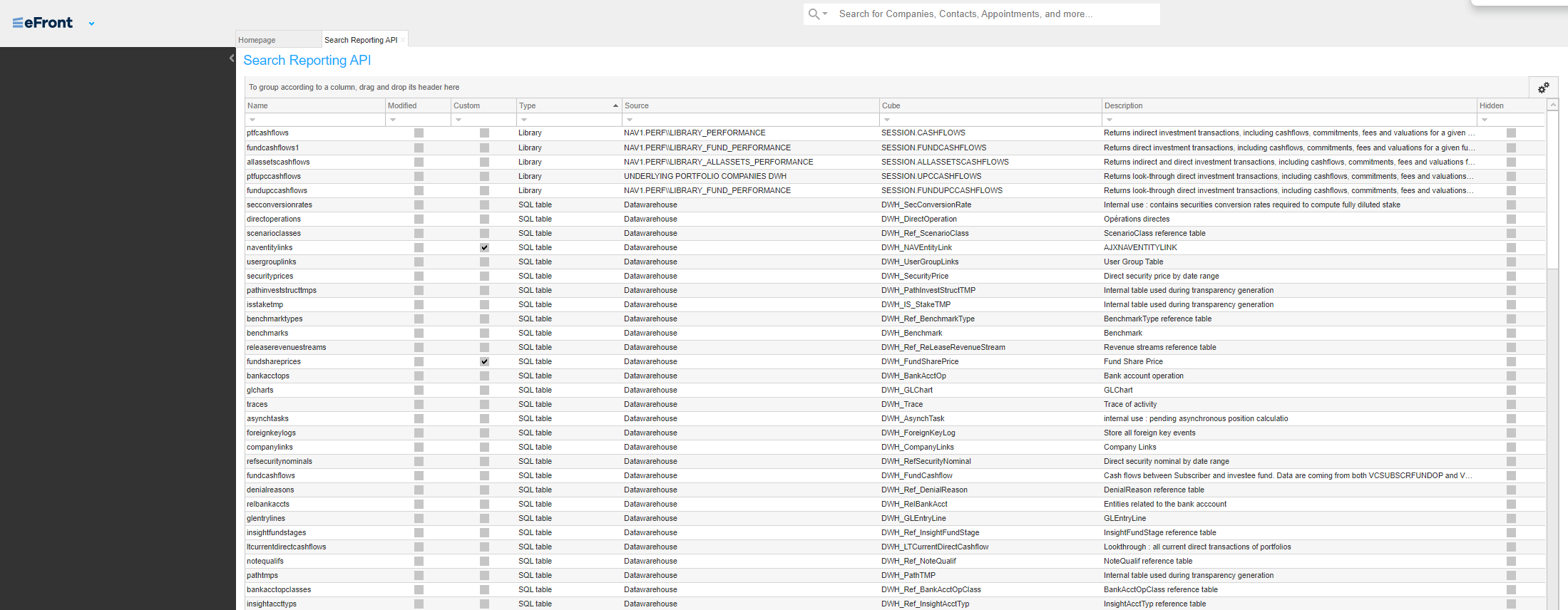
1. Check if Reporting API option is available under Administration. If it is not available, follow the next step to add Reporting API option in the menu.



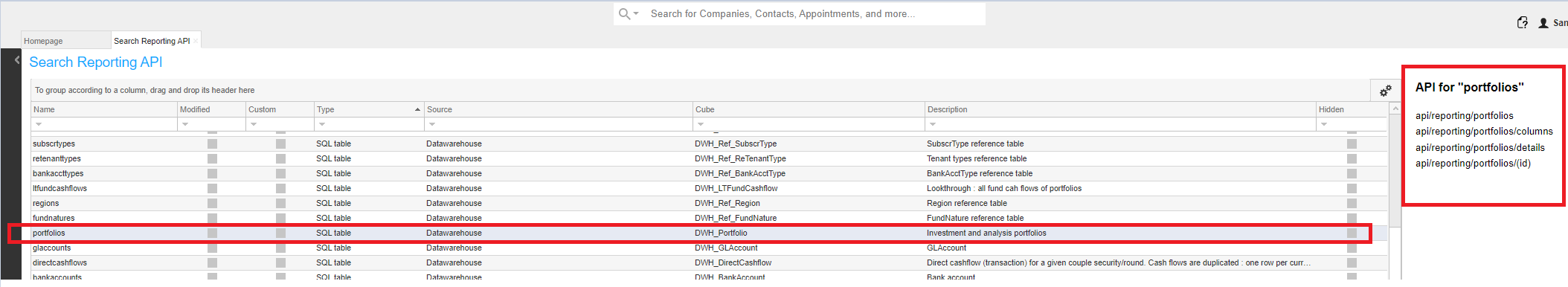
1. Go to Administration->Menus and select ADMIN menu in the Menus section of the page using Admin profile. In the Menu Items section click on Add button and add ‘Reporting API’ with entering {ADM\_DATAAPI} in Path or Expression column. Click on Save.



1. Reload the application.
2. From an administrative profile, navigate to the main menu and select **'Administration' -> ‘Reporting API’**. Once there you will get a list off all available API reports.



* For each report you select, endpoints will appear to the right of the table



api/reporting/reportapiname -> calls the API report

api/reporting/reportapiname/column -> returns the columns of the report

api/reporting/reportapiname/details -> returns the details of the report

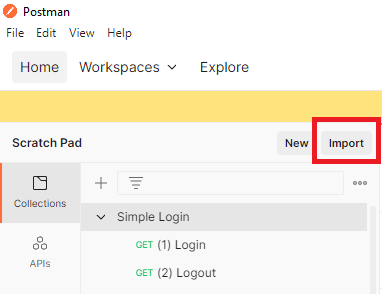
api/reporting/reportapiname/(id) -> returns data for 1 specific ID

# How to test and validate

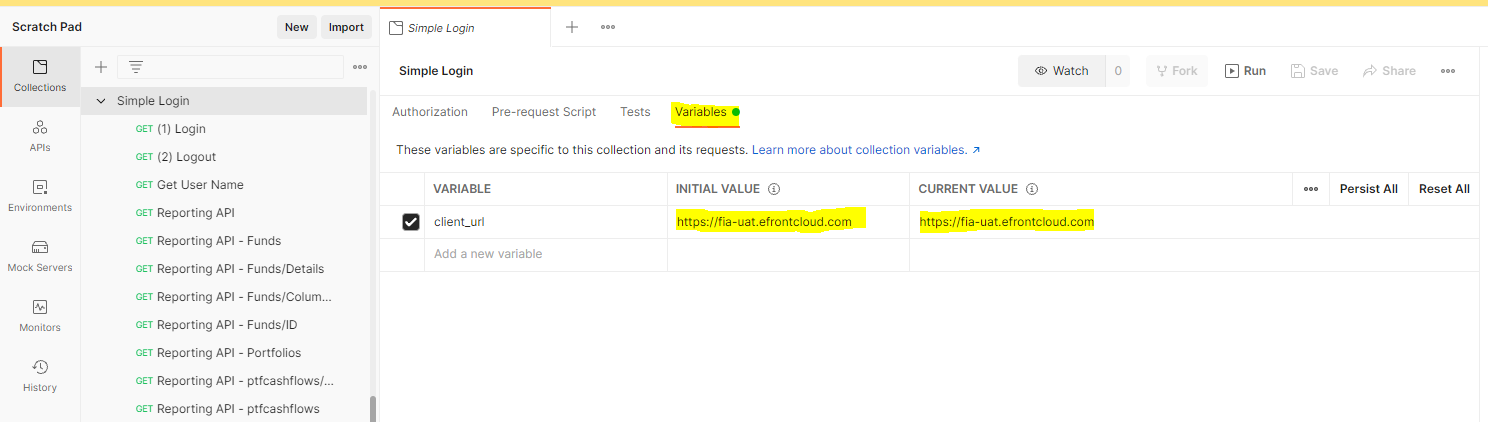
## Postman

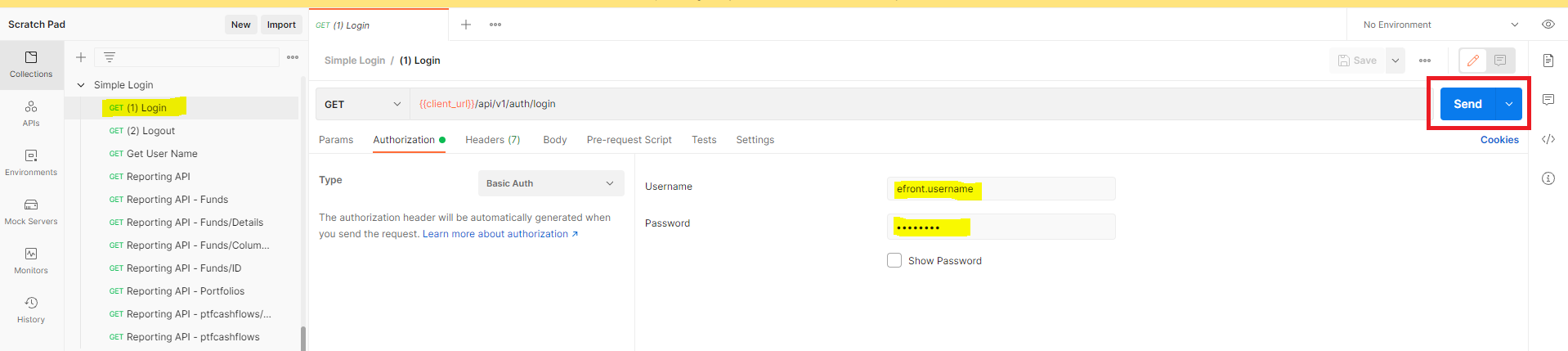
1. Import the provided Postman collection



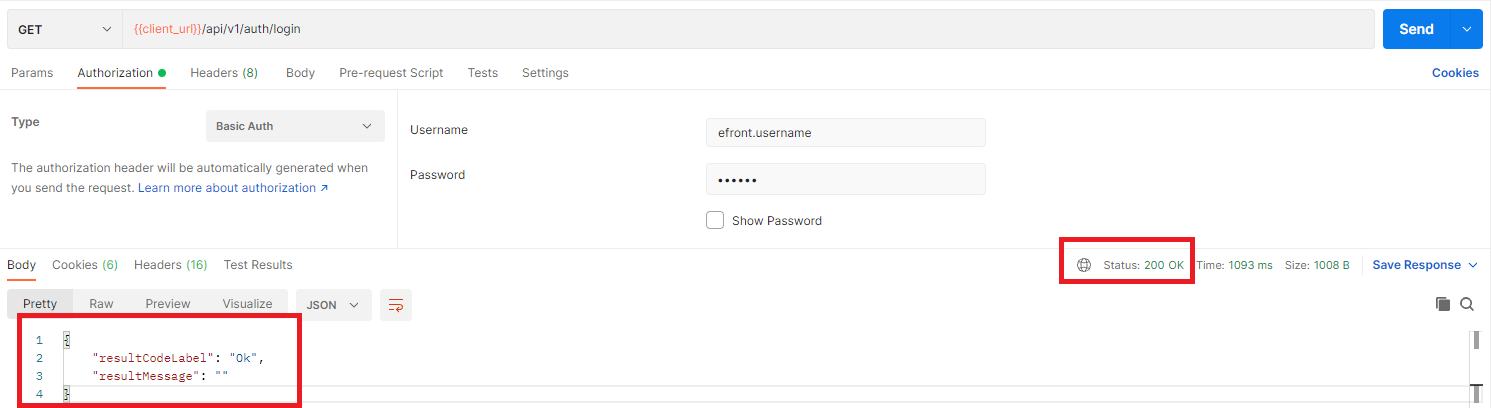


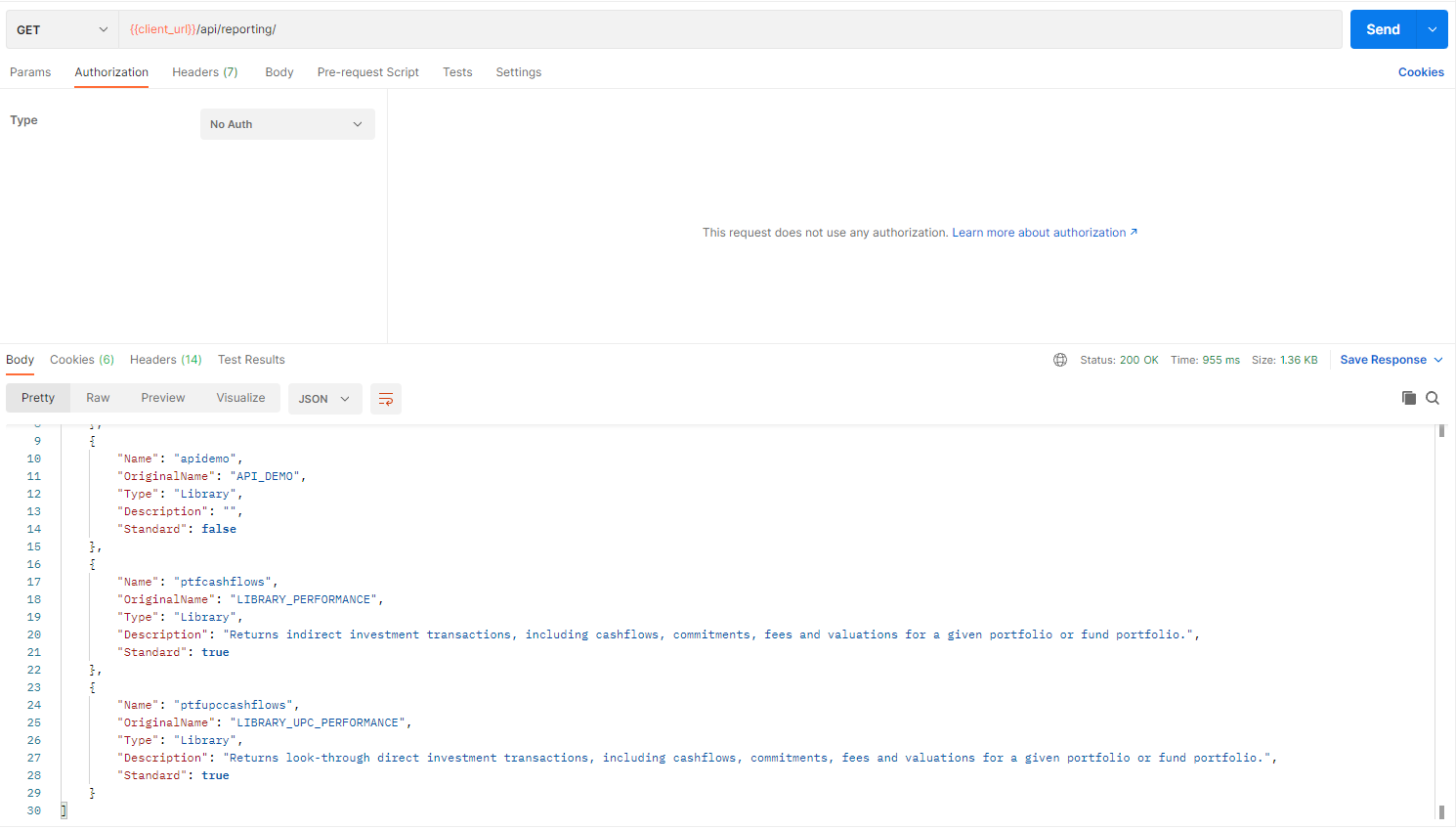
1. Click on the package and in the Variables tab, enter your application URL



1. In the (1) Login, under the Authorization tab enter your username and password  
   
2. If your connection is successful, you will get the following response 200 OK

And the following message:



1. To call an API just add the report name in the path and click send and you will see the results below:  
   e.g. **API/reporting** >> this list all available API’s  
   

* Note: If you get for the status code 401 Unauthorized it means your session has expired and you need to run again Invest Login request.

## Python

Sample Python script for connecting to and calling the reporting APIs.  
To use the script update the “Credentials” section accordingly.

import pandas

import requests

import json

import base64

# Credentials

rooturl = "https://fia.frontsrv.com" # FIA url

user = 'Username' # FIA Username

password = 'Password' # FIA Password

proxies = {} # define proxy if needed

payload = {} # Leave blank

UserPas = base64.b64encode((user + ':' + password).encode('ascii')).decode()

headers = {'Authorization': 'Basic %s' % UserPas}

session = requests.Session()

# 1. Login to Invest =================================================================

apipath = '/api/v1/auth/login'

print('> Login to %s...' % rooturl)

response = session.get(rooturl + apipath, headers=headers, data=payload, proxies=proxies)

print(response)

print(response.text)

# 2. Call the API report =============================================================

apipath = "/api/reporting"

print('> Calling %s...' % apipath)

response = session.get(rooturl + apipath, headers=headers, data=payload, proxies=proxies)

print(response)

print(response.text)

# 3. Transorm the response into a table to be used in powerBI ========================

data = json.dumps(json.loads(response.text))

sampleTable = pandas.read\_json(data)

print(sampleTable)