

GK REST API on Postman collection

Export: 2022-03-25

COPYRIGHT

© GK Software Group 2022. All rights reserved.

These materials are provided by GK Software SE and/or its affiliated companies (GK Software Group) for informational purposes only without representation or warranty of any kind, and GK Software Group shall not be liable for errors or omissions with respect to the materials.

The only warranties provided by GK Software Group are those that are set forth in the express warranty statements in the individual agreements between GK Software Group and its Clients or Partners. Nothing herein should be construed as constituting an additional warranty.

No part of these materials shall be reproduced or transmitted to third parties in any form or for any purpose without the express permission of GK Software Group. The information contained herein may be changed without prior notice.

All company names of GK Software Group, GK Software Group's product/services names as well GK Software Groups logos mentioned herein are registered trademarks and intellectual property of GK Software Group.

Internal Document Information: 1096025770 | 2022-03-25

TABLE OF CONTENTS

1	Cookbook Notes	4
1.1	Table of Contents	4
1.2	Cookbook preface.....	4
1.3	Typographic conventions	4
2	Postman introduction	4
2.1	Postman description	4
2.2	Postman download.....	4
2.3	Postman installation.....	6
3	Setting up development environment	6
3.1	Downloading the Swagger documentation	6
3.2	Postman collections.....	7
3.3	Postman environment	10
4	Postman features	12
4.1	Collections features.....	12
5	Omnibasket	16
5.1	Usage of Omnibasket.com.....	16
5.2	Service API	16
5.3	App-Enablement.....	17
6	How to use the REST API	17
6.1	Complete Documentation	17
6.2	Example Request	19

1 Cookbook Notes

1.1 Table of Contents

1.2 Cookbook preface

The ultimate goal of this book is to give you answers to all your questions related to the Service API or GK REST API and how it can be used on the available Postman collection.

1.3 Typographic conventions

For easier reading and better understanding, we will use the following typographic conventions and symbols in this documentation.

Format	Description
bold	References to the screen display: Words or characters quoted from the screen. This includes field names, pushbutton names, menu names, masks, menu paths and menu options. Example: Select OK.
<i>bold and italic</i>	Highlighting: Highlighted words or expressions in the body text. Example: Please always follow the instructions.
Proportional spacing bold	Exact user input: This includes words or characters that must be entered by the user in the application exactly as indicated in the documentation. Example: Amount: 20.00 EUR
CAPITAL LETTERS AND BOLD	Names of keyboard keys: Example: SHIFT, CTRL, ALT

2 Postman introduction

2.1 Postman description

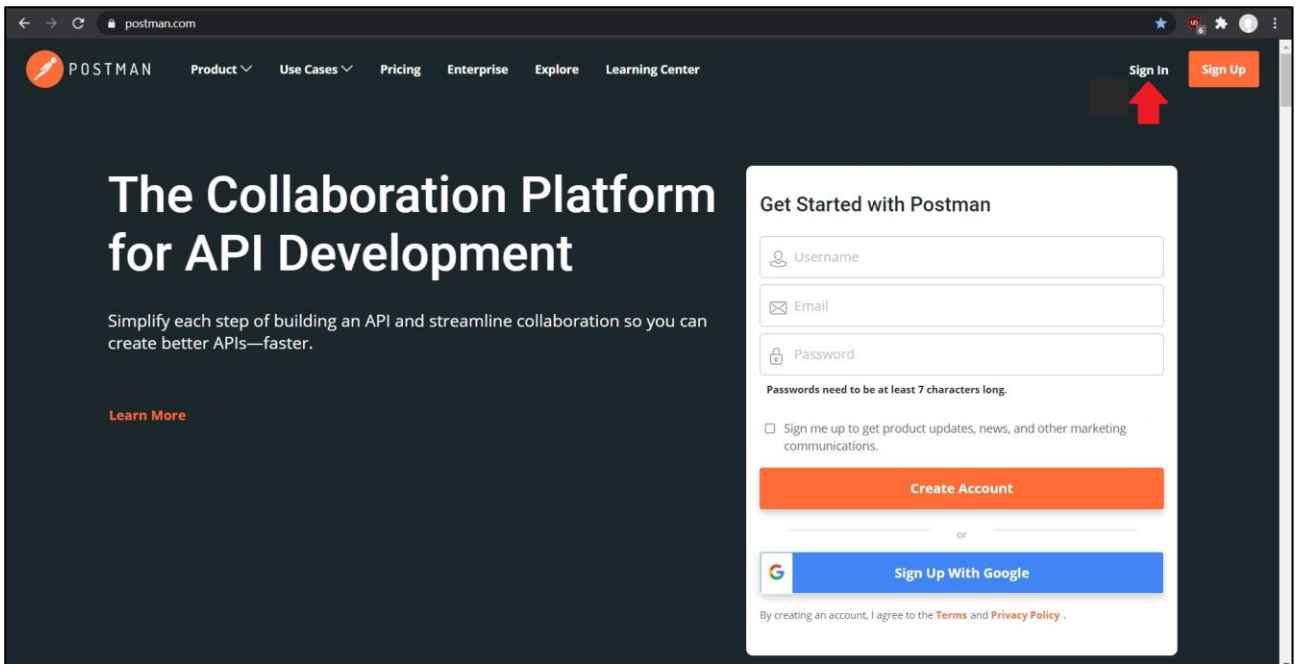
Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster. This includes developing, testing, sharing and documenting your APIs. Postman also gives to possibility to manage your API data via Postman's desktop, web, CLI, and/or API interfaces.

2.2 Postman download

There are two ways how to use the Postman API Platform. You can either download it from their official website or use the web version in your browser.

<https://www.postman.com/>

Before you start with Postman, you should first register on their website. It is necessary to activate your account by confirming your mail address. If you are done with that, **refresh the Postman website** and sign in by **clicking the sign in button**.

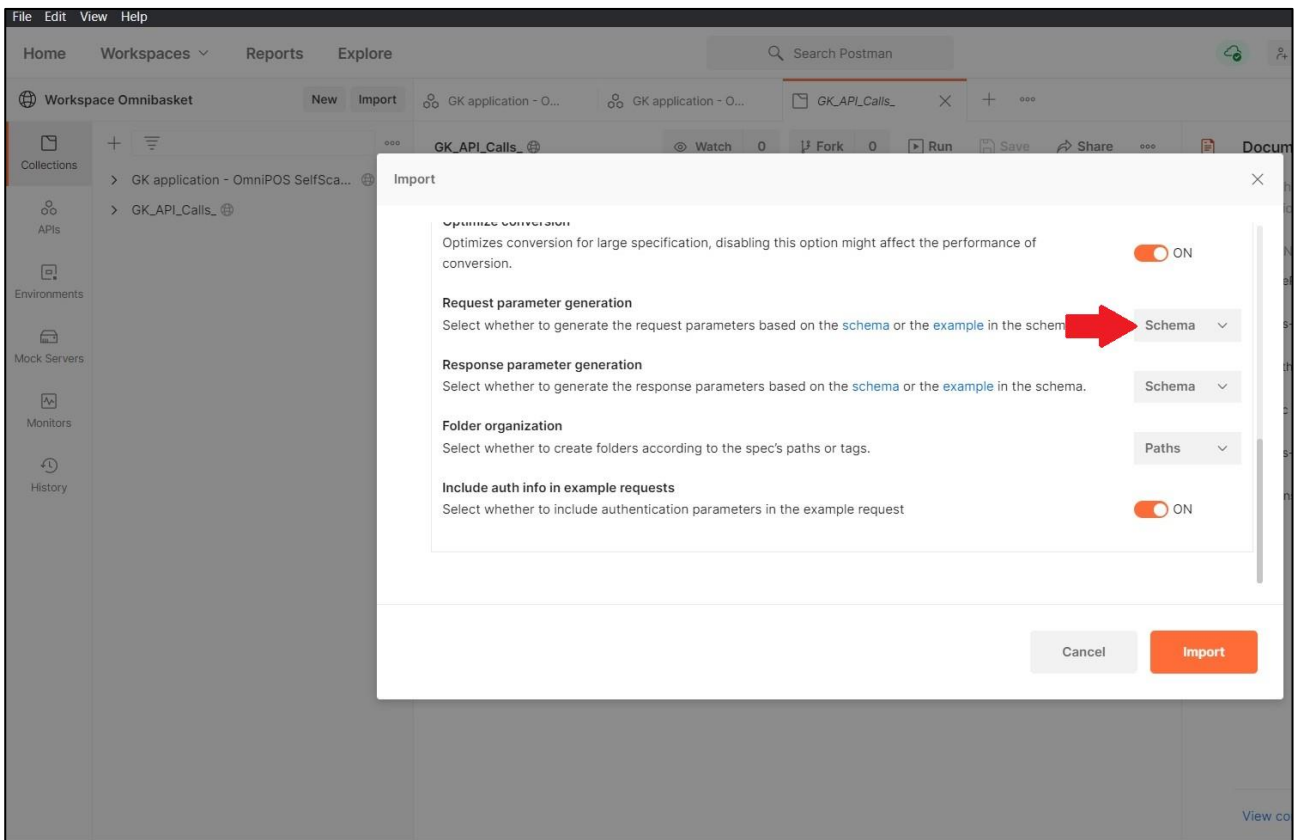


Now the Postman web application should start. If not you have to **click the Launch Postman button** if you want to use the Postman web version.

But I would recommend using the Postman application, because it gives a full-featured version of Postman. Postman application must be version 9 and higher related to new security functions.

<https://www.postman.com/downloads/>

Click Download the App and select whether you have a 32-Bit or 64-Bit windows system. Now you are downloading a .exe application file which helps to install Postman. The name of the app should look somewhat like this "Postman-win64-9.9.3-Setup.exe".



2.3 Postman installation

Open the downloaded file by **double clicking on it**. Now a new window opens where you can just **click on sign in**. This button is a link to their website where you should be able to sign in and verify your Postman application. Now you should have the full Postman version running on your system.

3 Setting up development environment

3.1 Downloading the Swagger documentation

First of all use this link to register on the Omnibasket platform. The registration process is a key thing to do, to access all information on Omnibasket.com. The process could take some time.

<https://omnibasket.com/>

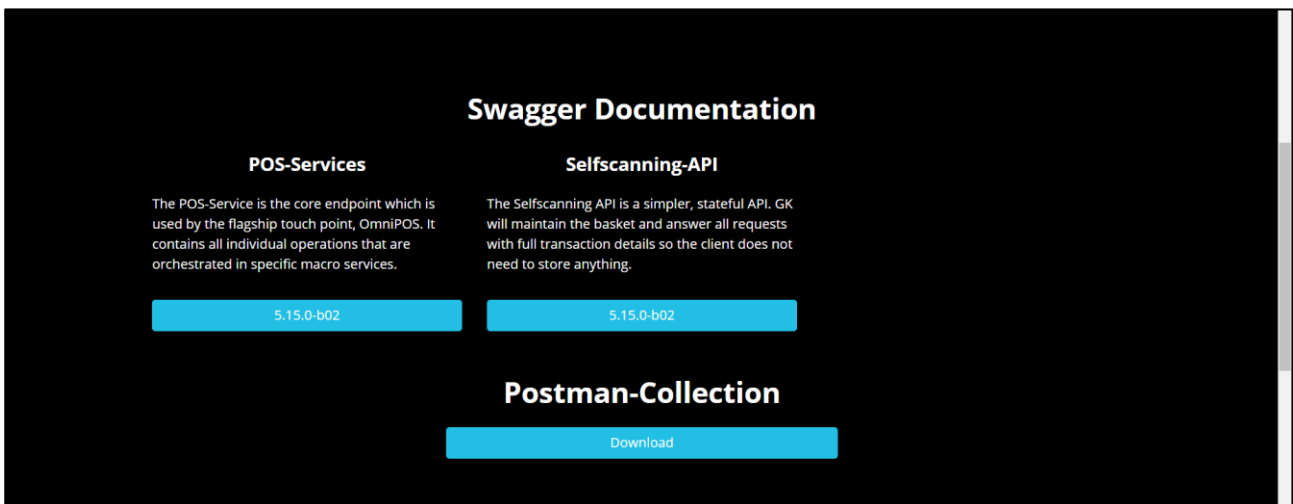
To gain a sandbox developer access you will need to:

- **Click the registration button** below, provide contact information, company info and proposed solution
- Once approved you will gain access to a webservices public sandbox.
If you wish to use AppEnablement- further information about the use case will be required. We will probably invite you to a quick chat, and for selected use cases will activate a Test & Try advanced environment for you to try your idea out.
- To be added to our catalogue and be available in production: commercial agreement and a small certification effort will be required

After your account is registered and verified you can go on

<https://omnibasket.com/service-api>

and download the Swagger Documentation of the POS-Service and the Selfscanning-API. The buttons for the files should be named like 5.15.0-b02 depending on which version they currently are. Those three files are now in your download space on your PC. It's helpful to move them into a directory where you save all your files who have something to do with the Omnibasket and Postman.

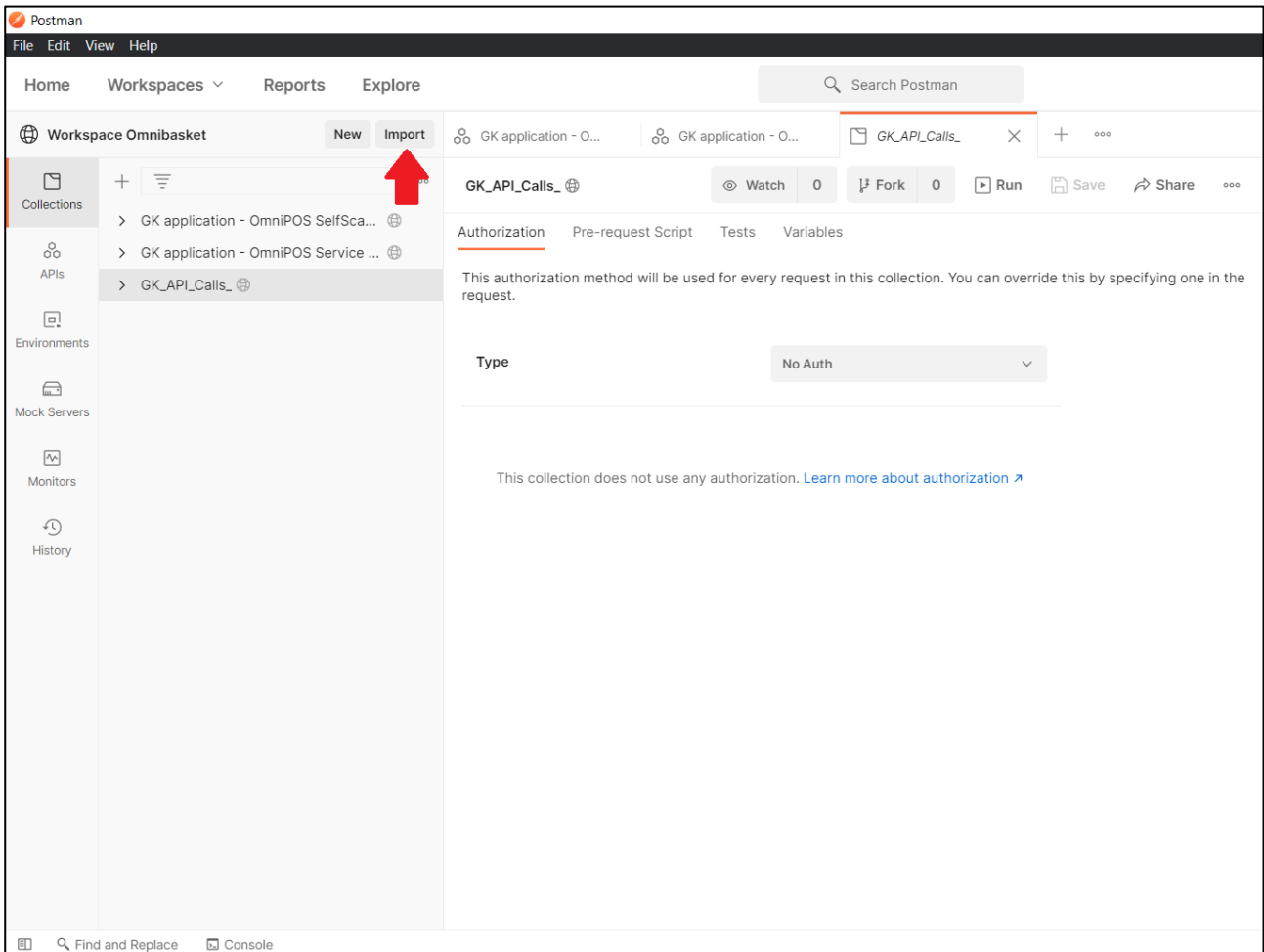


3.2 Postman collections

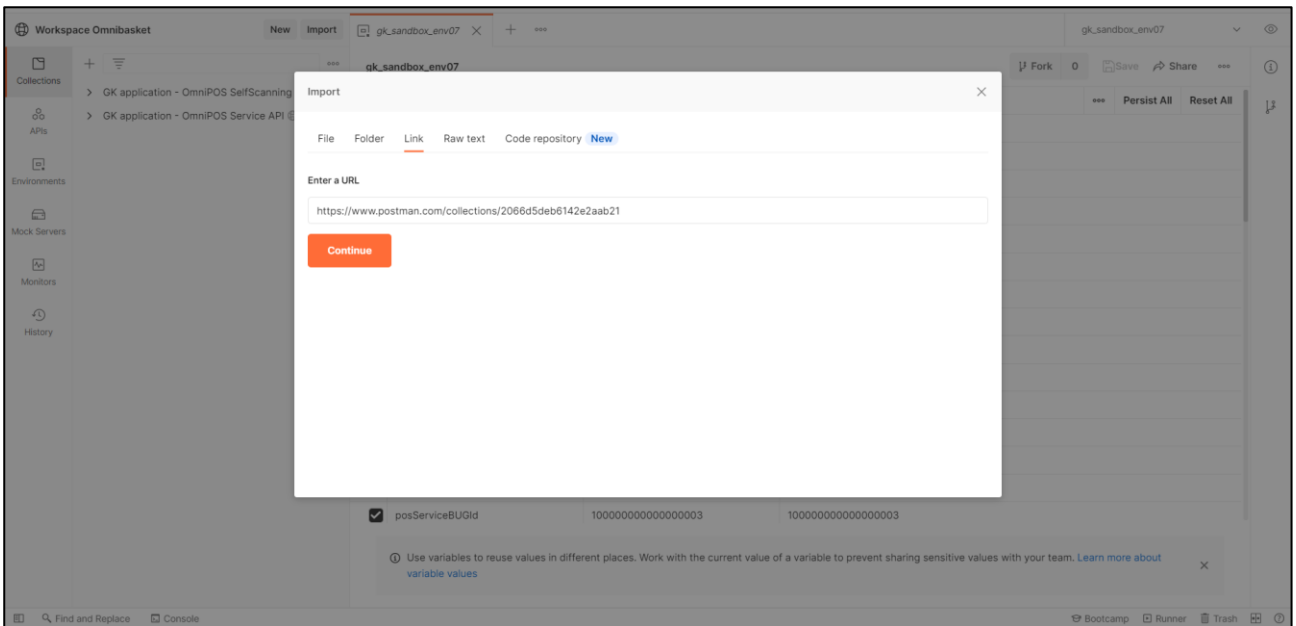
After you successfully downloaded the Swagger Documentation, you should **start** the previously downloaded Postman Application where you can sign in with your account.

First of all you should create a new workspace in Postman. You can do this by **clicking the new button** and selecting Workspace. For the reason that this workspace will keep all your GK Omnibasket related Collections you shall call it *Workspace Omnibasket*.

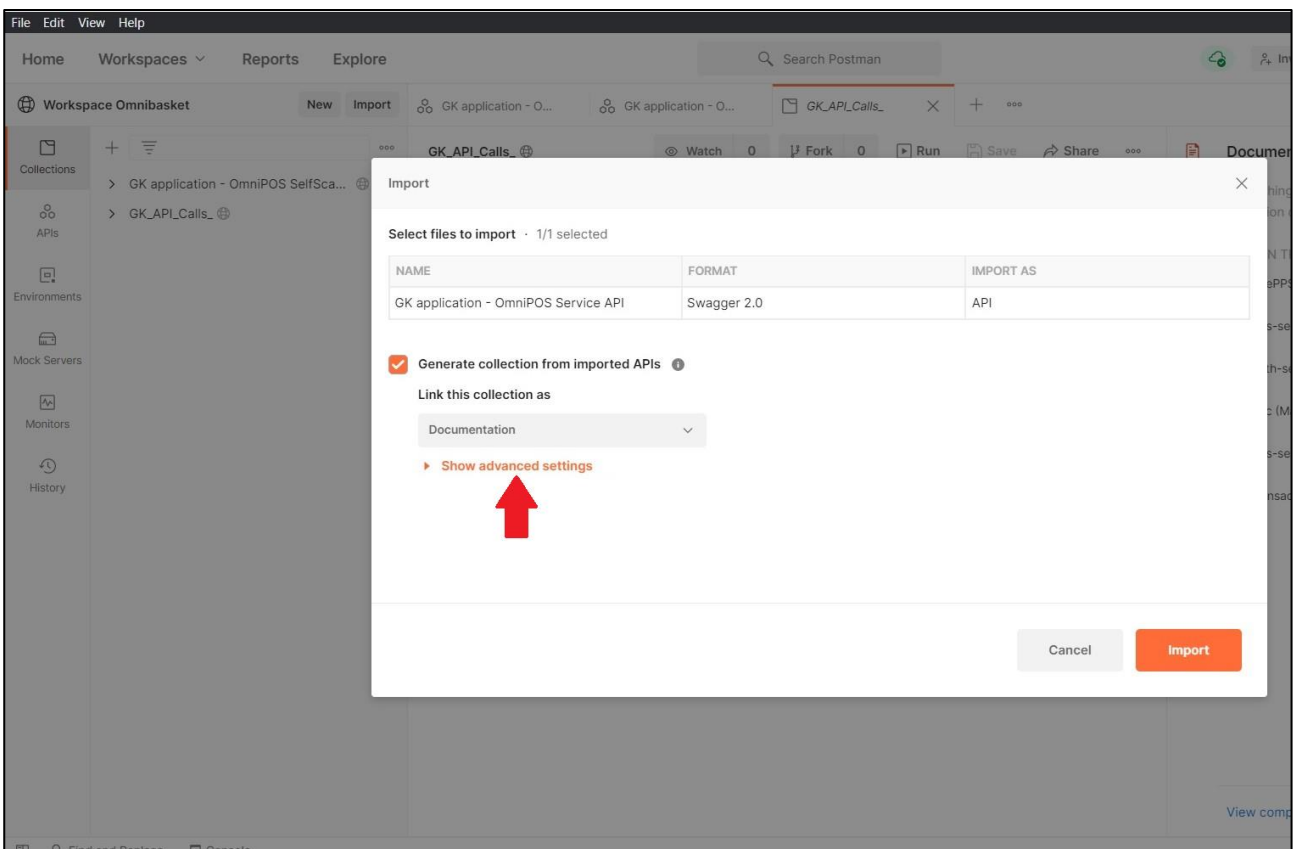
To download the Postman-Collection you will have to go on omnibasket.com/service-api again and **click the Download button** below Postman-Collection. A website will show up where you can just copy the URL. Than go in t Postman and in the Workspace Omnibasket where you can **click on the import button**



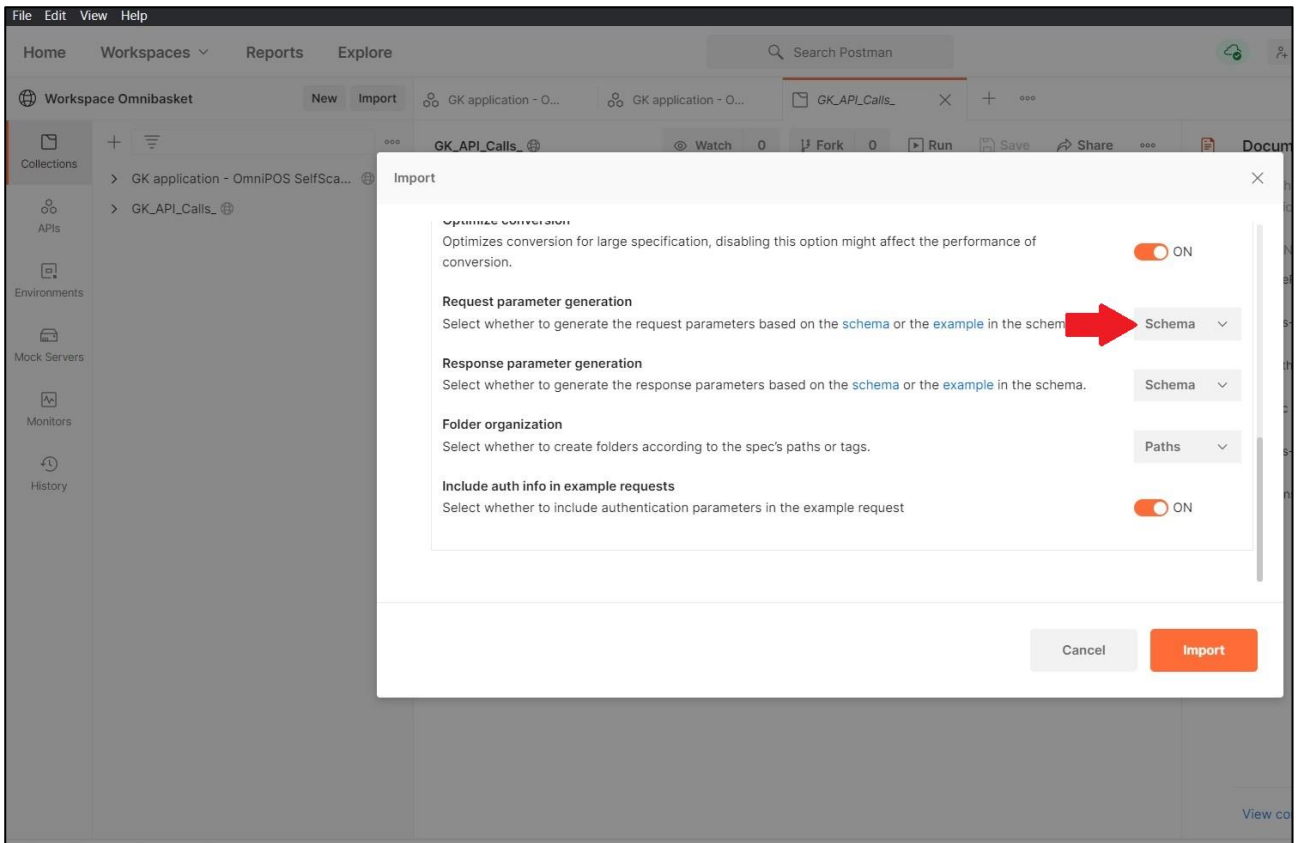
and **paste the Link** in the Link section. The installation will start after you confirm it and the GK_API_Calls collection should show up in the collection section.



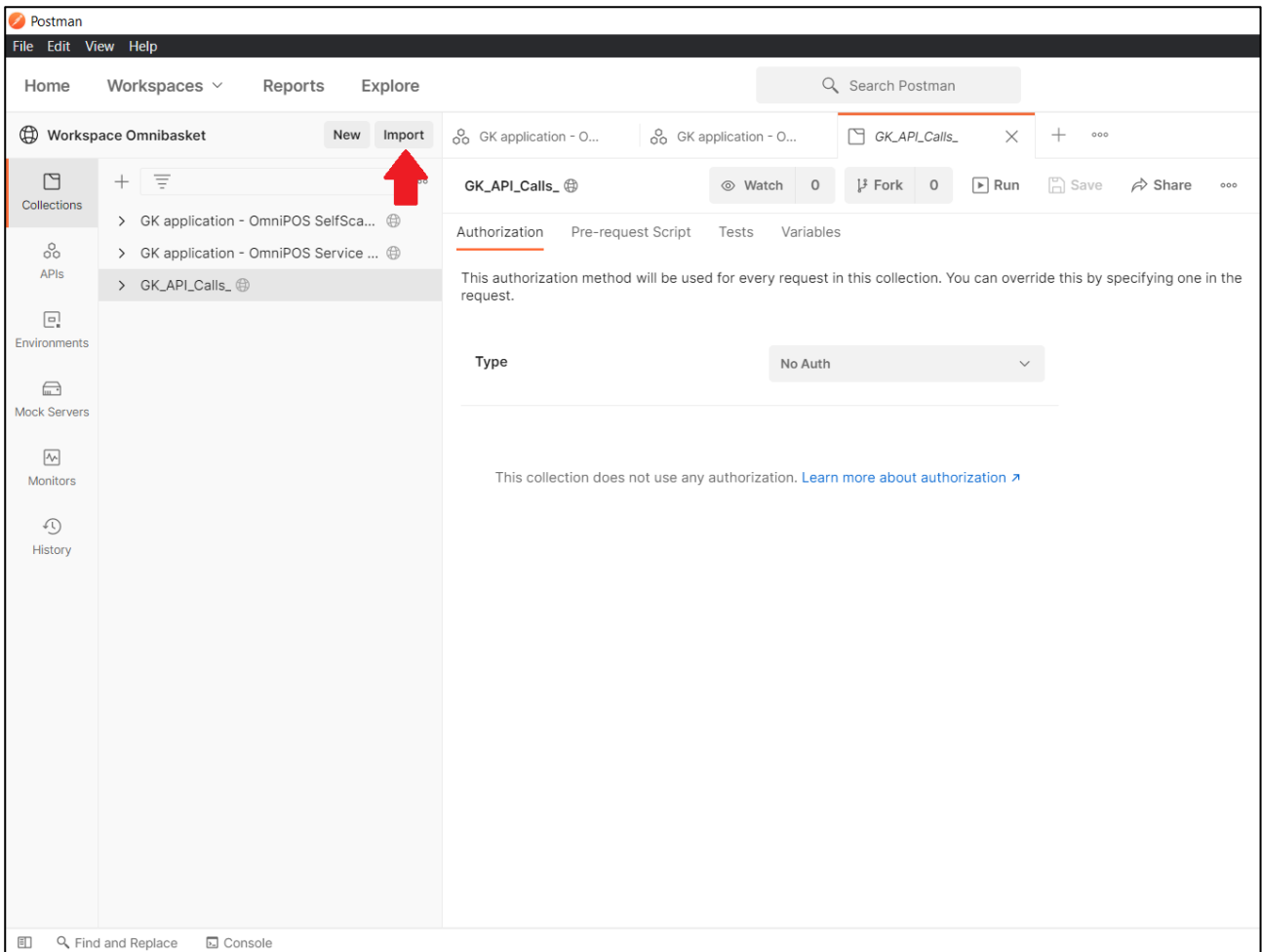
To add the Swagger Documentation you have to **click on Import** again and **select** in the section file menu the button **Upload Files**. You should start by adding the GK application - OmniPOS SelfScanning Service API. It should show up in your Collection menu. For the second one, the GK application - OmniPOS Service API you will have to **click on show advanced settings** after you selected the correct file.



Now make sure that Request parameter generation and Response parameter generation is set to schema. If so you can start the import.

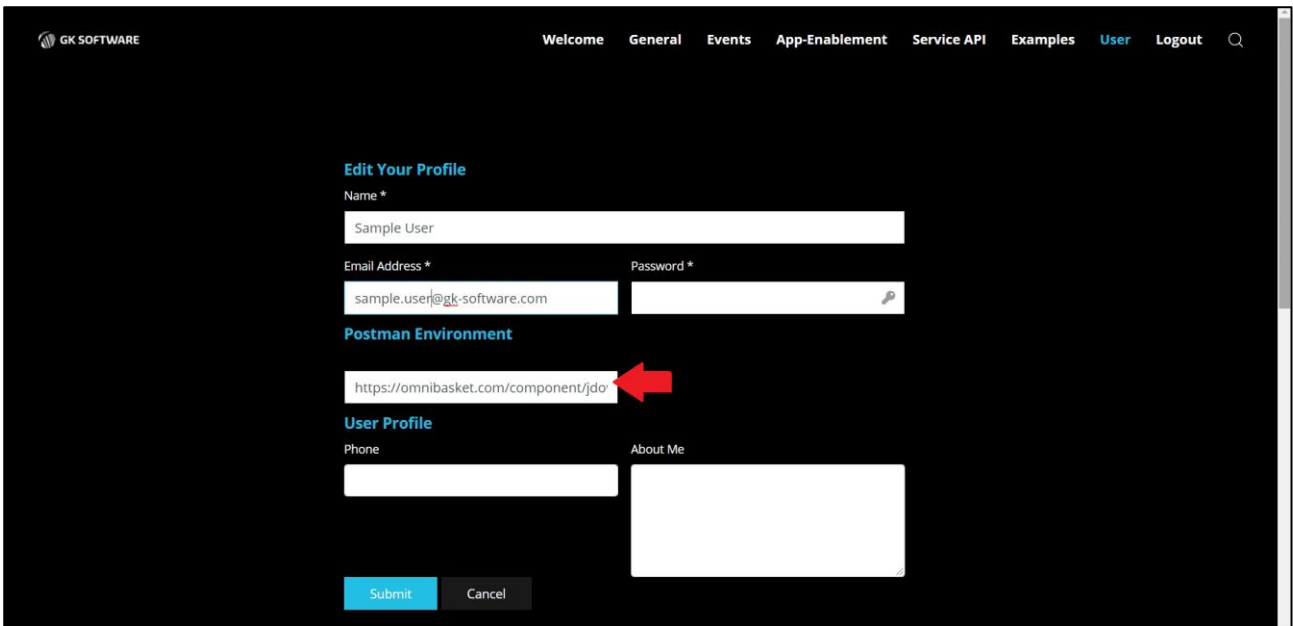


The collections menu inside your Omnibasket workspace should now look like this.

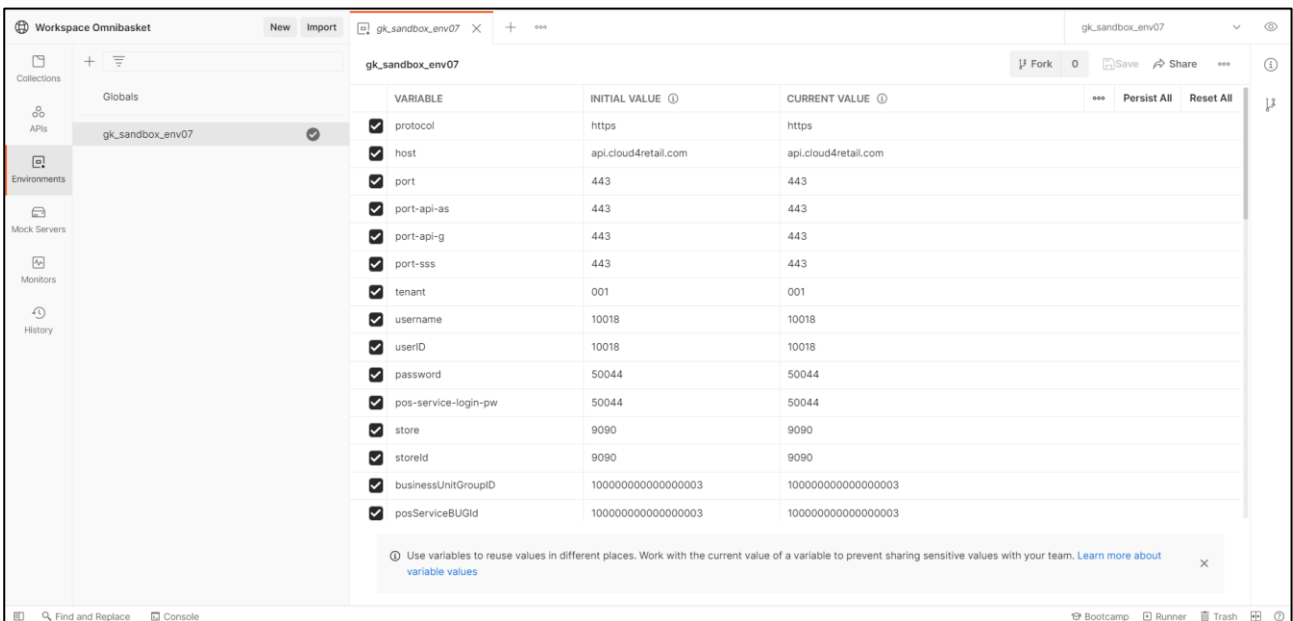


3.3 Postman environment

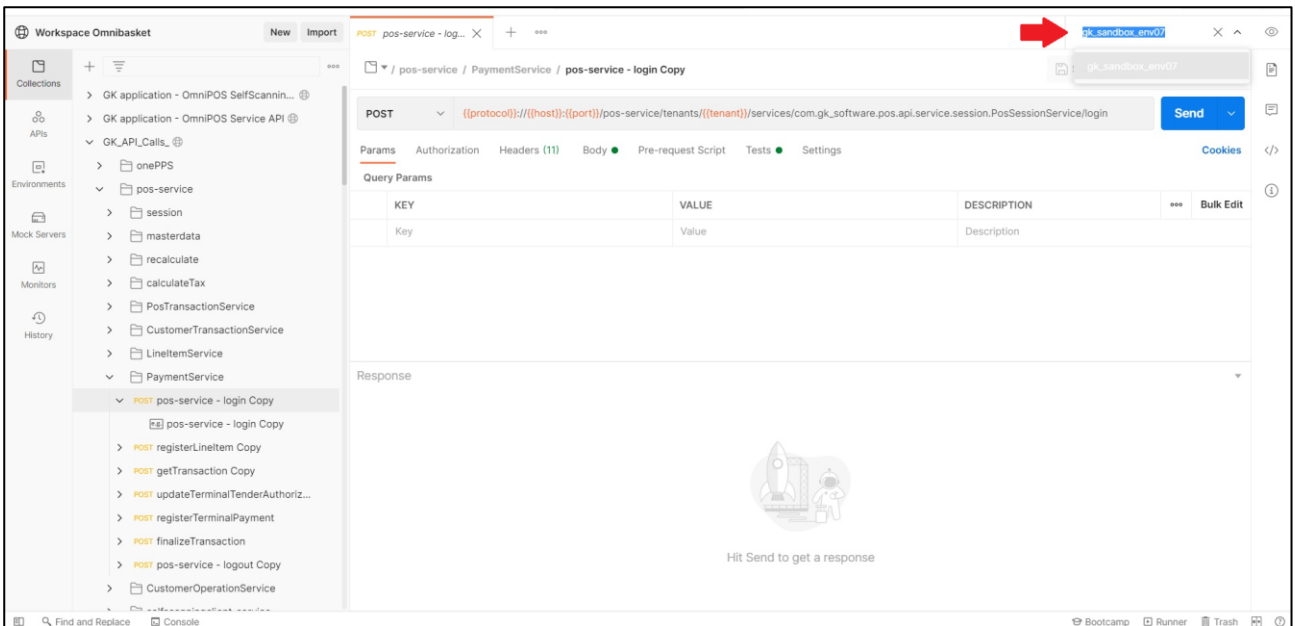
Right now you are not able to send any request and get an answer, because they key variables aren't set yet. Therefore we need an environment. You should already have the necessary .json file, because you are supposed to get it after registering on Omnibasket.com. You can download it by going onto Omnibasket.com and going in the user section. After you logged in, a URL should show up under the Postman Environment point. **Copy this link** and paste it in any browser. The download will start automatically.



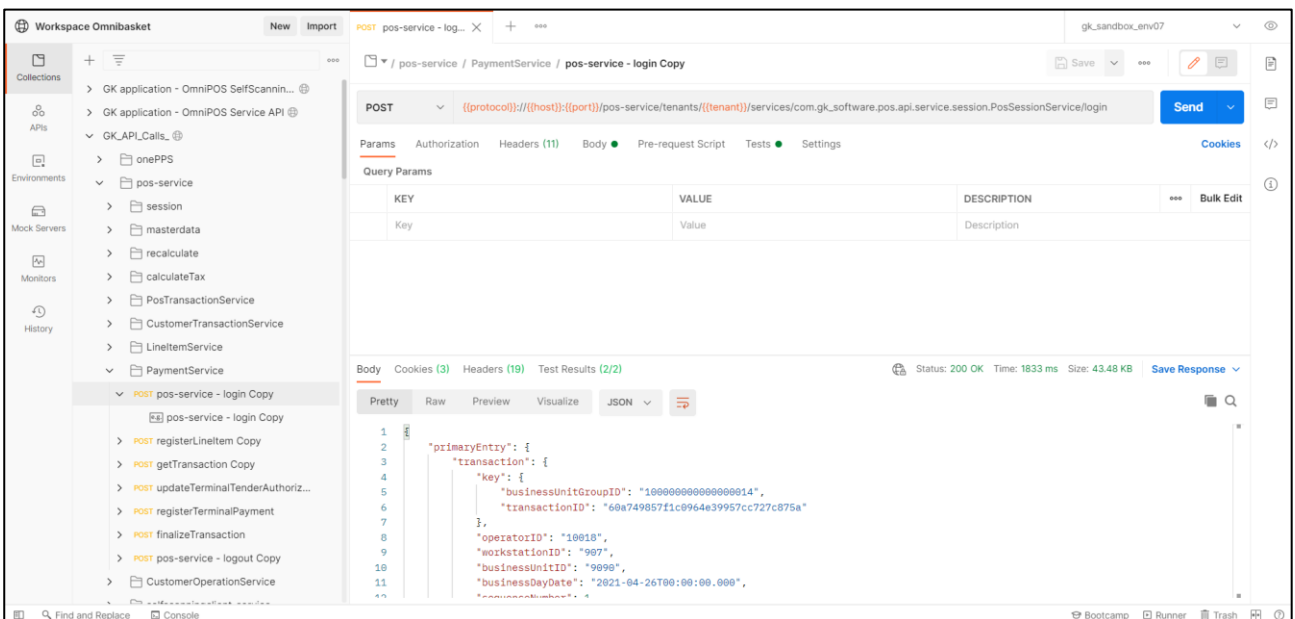
You can install your environment into Postman by going in the environment section and again **clicking the import button** and **selecting the correct file**. Now the .json file should show up in the environment section.



You are still not able to send any request. To fix that you have to go in the collection section and **click on API Calls**. In the top right corner you can select the environment what you want to use. **Change it** to the previously installed one.



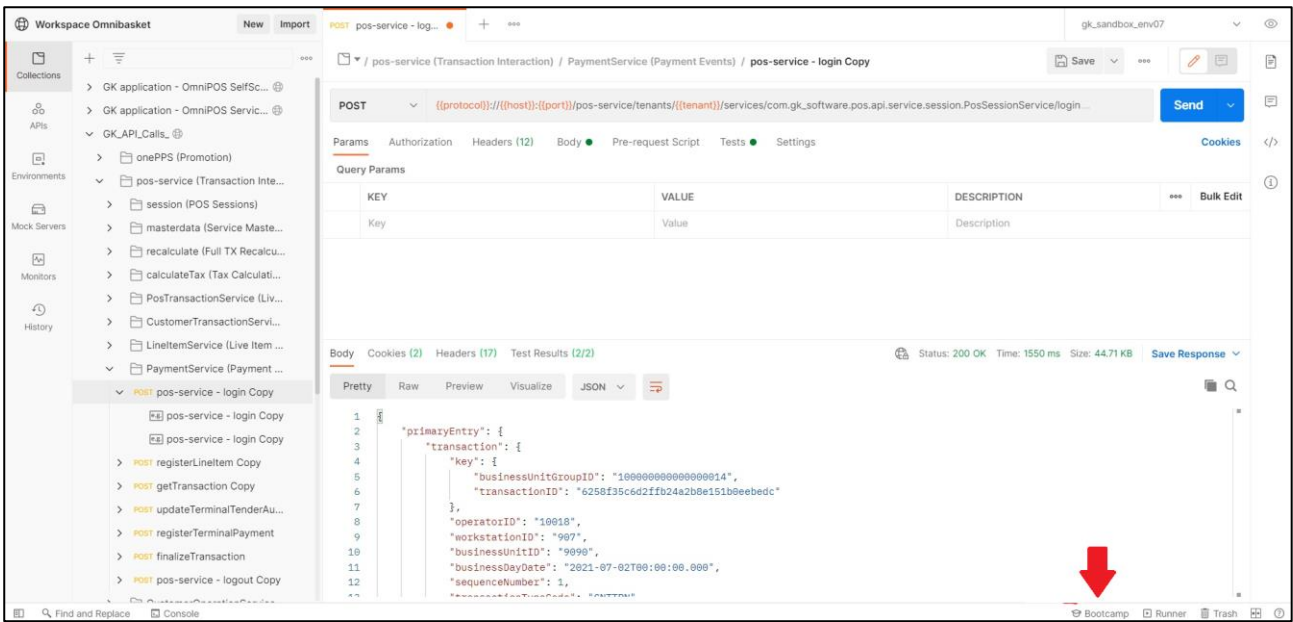
Now you should be able to send a request. Checked if all works by sending a request in the PaymentService/pos-service folder using the pos-service login method. Which gets you with the following response.



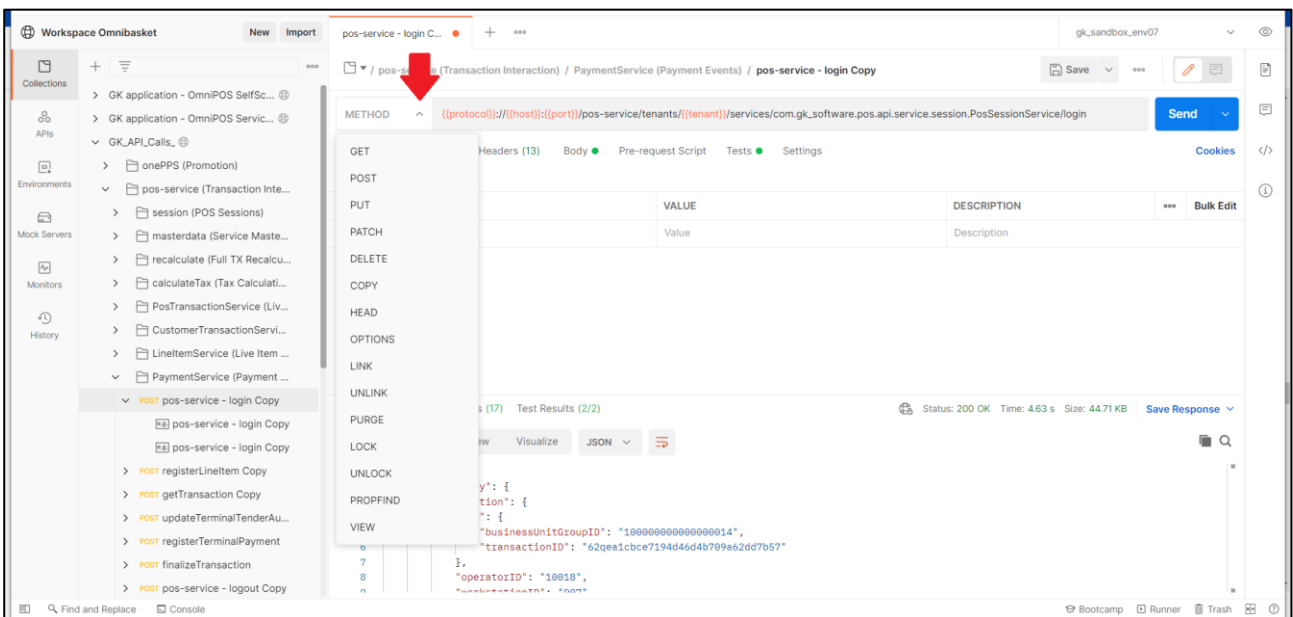
4 Postman features

4.1 Collections features

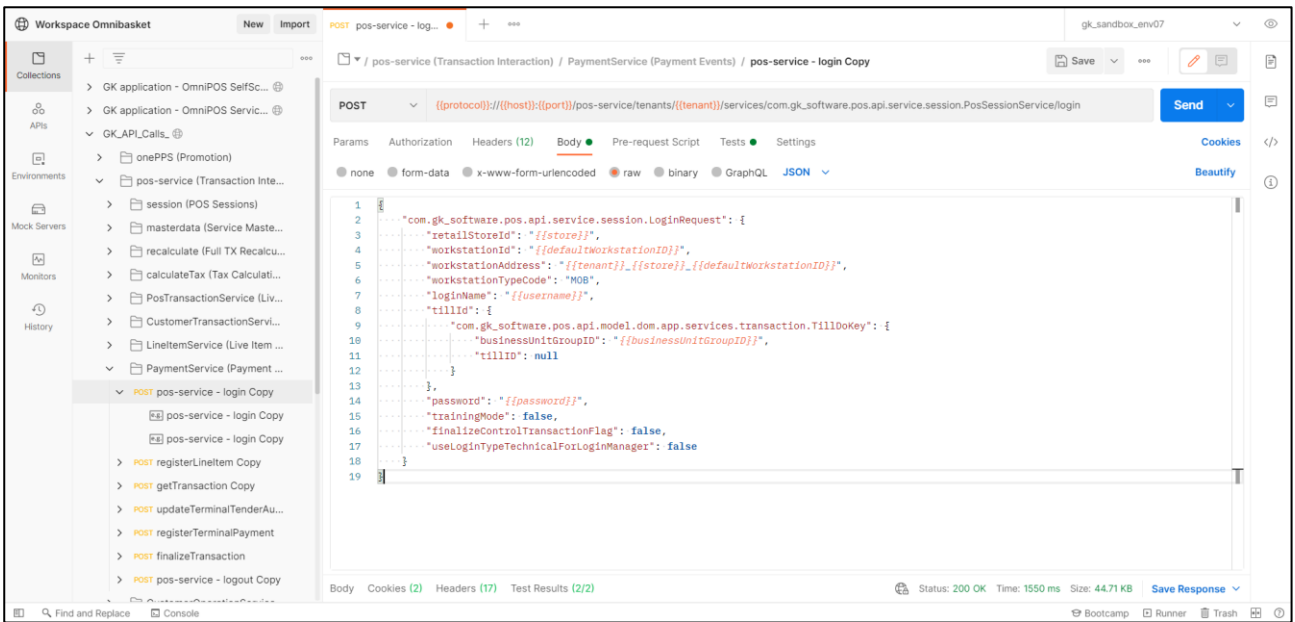
Postman is a very complex and well thought out tool. Therefore it has a huge number of features and support tools. You can read about all Postman features right here <https://www.postman.com/postman-features/>. It is also helpful to have a look at the Postman bootcamp in the Postman application. To get there, you will just have to **click on Bootcamp** in the bottom right corner, on any open Postman site.



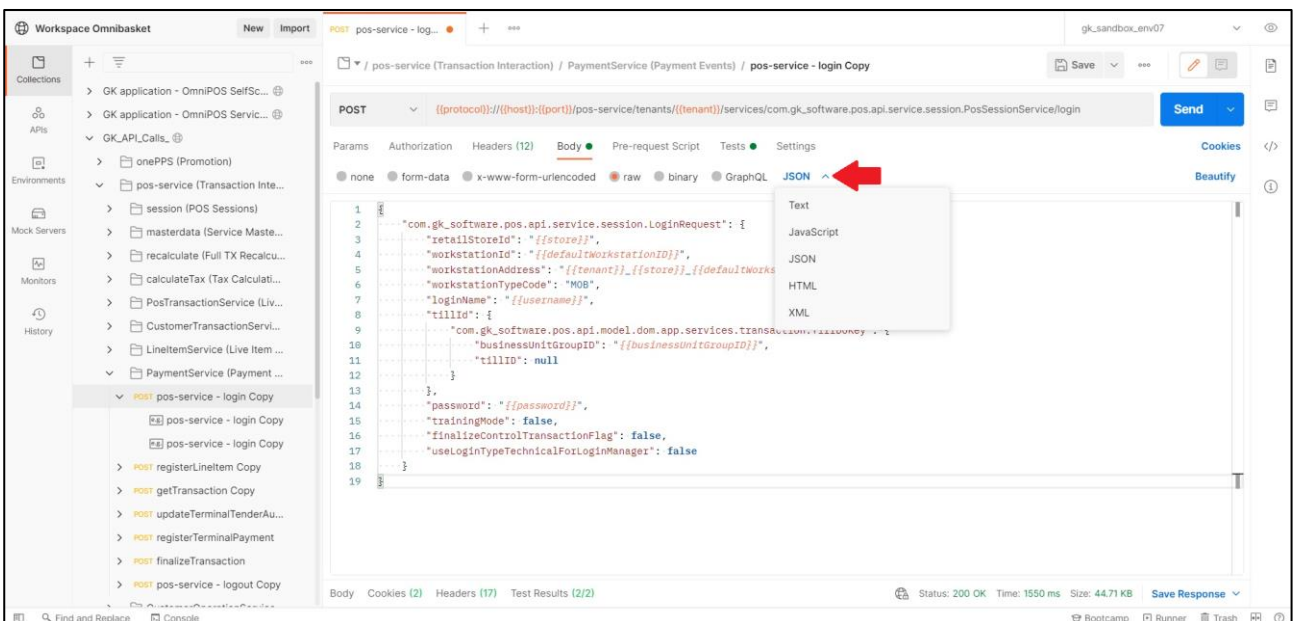
Nice to know is, that every request has a specific type like POST or PUT or GET. There is a drop down list of all types if you **click on POST** in front of the requests name. The different types are important because they show and decide, what every request is used for.



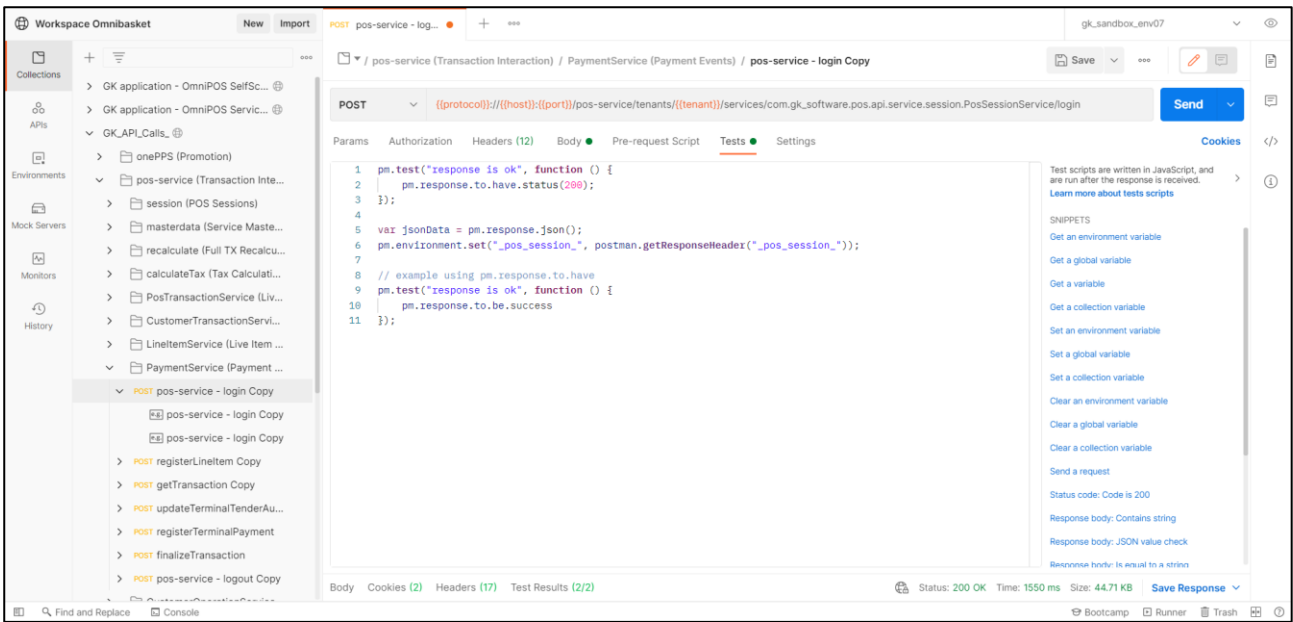
Now let's have a look at the most important features you are using during your work with postman. In the row of sub items below the name of the request you can find seven fields. Params, Authorization, Headers, Body, Pre-request Script, Tests and Settings. First of all we look at the body. In there you can see the entire performed code. Variables which are colored in orange refer to parts of the API.



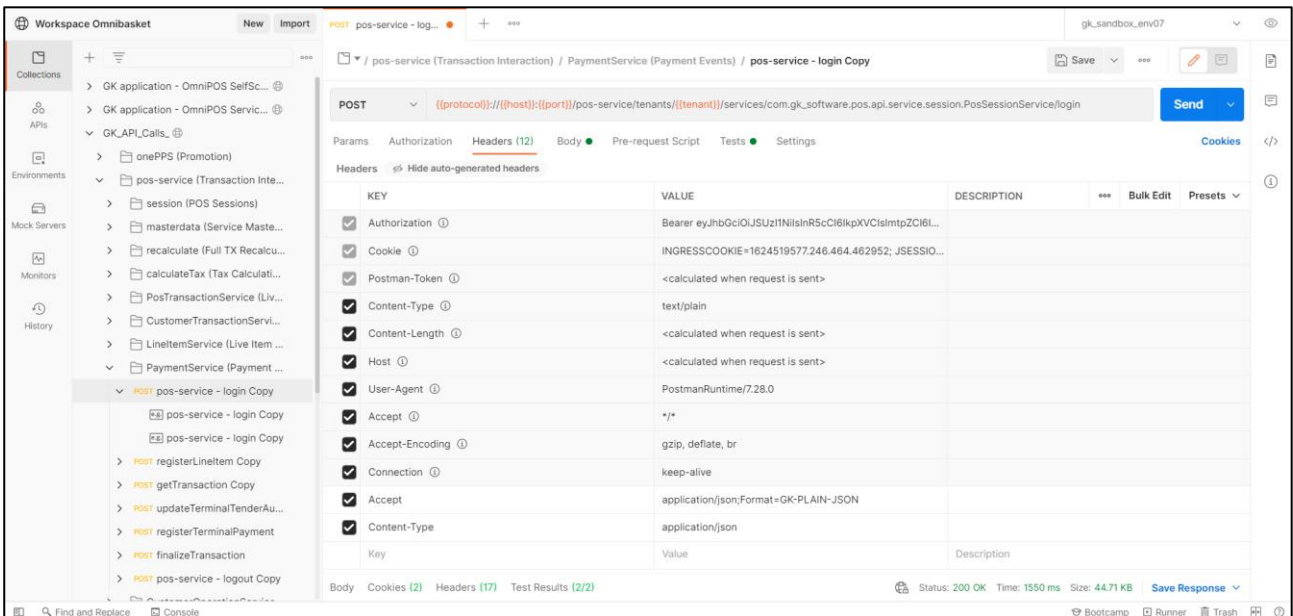
You can change the type of representing the request, by **clicking on JSON** and selecting another type in the drop down list.



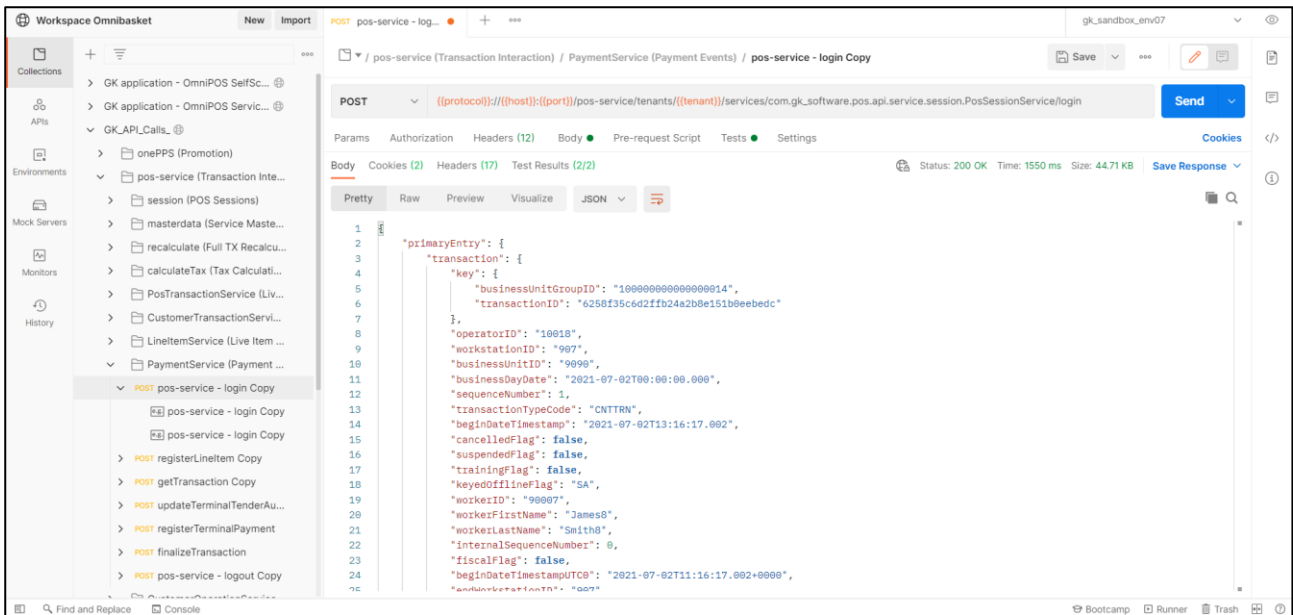
The next sub item we want to talk about are the tests. Sometimes this is very useful, because the body of some requests could be quiet big and not very manageable, but in the tests section you will find the most needed methods of the request being tested.



Another important sub item is the headers section. It shows all needed information about what a request needs to run. If you do not know what each header does, you can just hover of the small info sign next to the header name.



Last but not least let's see what the request response actually shows you. In the top right you will always find the information, if the response worked as supposed. The status 200 just means it's all good, time is the response time in ms and size the response size in KB. There is also another Test Results sub item just for the response and you can also change the type of representing the response.



5 Omnibasket

5.1 Usage of Omnibasket.com

Omnibasket is the developer portal for accessing GK's retail platform, the platform that lets you bring the basket to your customer wherever they are.

<https://omnibasket.com/>

Here you can:

- Use GK's web services to integrate retail business logic to applications outside traditional POS -- mobile, web, smart cars, or any other customer touch-point
- Integrate web plugins to GK in-store POS platform using simple Javascript. Quickly bring recommendations, clienteling, time and attendance, or any other app you can imagine to the POS
- Discuss with other developers in the forum or get support from our experts
- Download and install a fully functional POS: mobile or stationary

Use the Portal, for information, Swagger & Postman doc's and samples of our REST APIs and AppEnablement. The two main ways you can use the platform for are:

- Use the remote platform web services to develop a new client or embed specific functions in your existing applications
- Use AppEnablement to develop web plug-ins for POS clients with the ability to communicate with the POS via a Javascript bridge

5.2 Service API

The Service API is your entry point to the GK retail platform.

- If you would like to develop brand new applications or embed GK functions in your existing app, then use the Service API
- Examples include a new consumer mobile client, or embedding our services in your existing consumer mobile app

- APIs cover functionality like security, basket management, basket calculation, promotions, loyalty, consumer management, user management, and more
- See documentation and samples for more info

POS-Service: The POS-Service is the core endpoint which is used by the flagship touch point, OmniPOS. It contains all individual operations that are orchestrated in specific macro services.

Selfscanning-API: The Selfscanning API is a simpler, stateful API. GK will maintain the basket and answer all requests with full transaction details so the client does not need to store anything.

5.3 App-Enablement

You can find and download a very useful documentation for the App-Enablement on <https://omnibasket.com/app-enablement>, if you scroll down and **click Download Documentation** below the 'How to start coding' header.

There is also a video of how one of our experts shows how we embed an example app into the POS, at the bottom of the page.

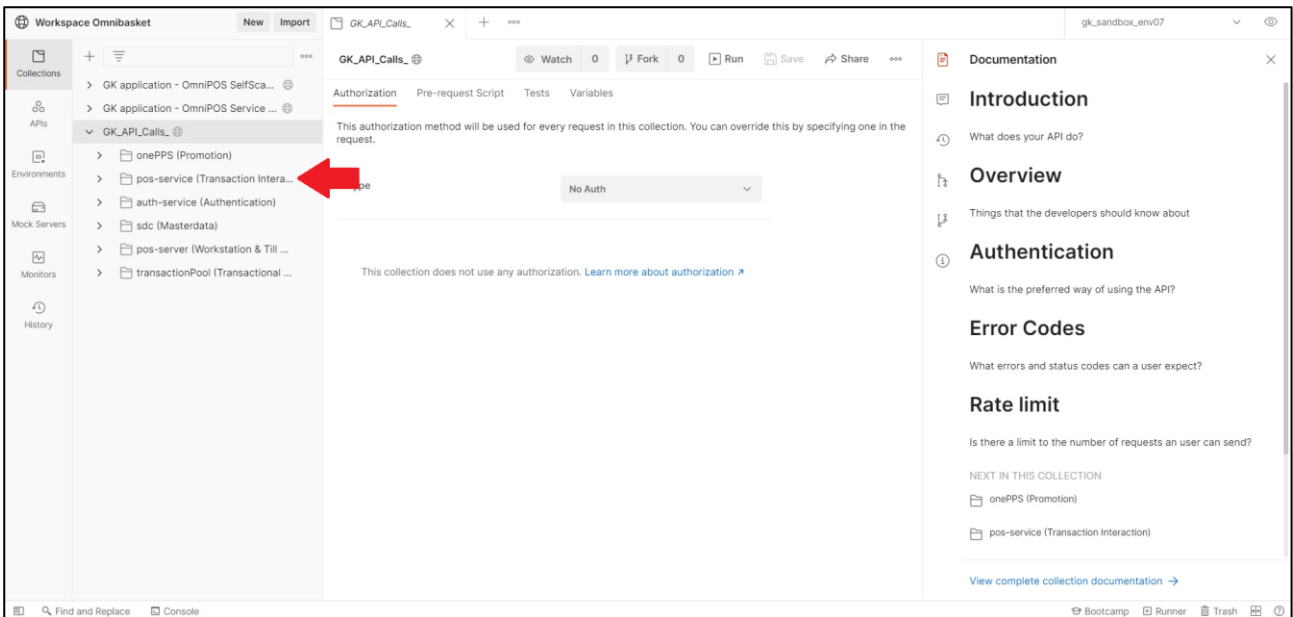
- App-Enablement is
- a plug-in interface that allows integrating web applications into the GK POS
- implemented as a JavaScript bridge enabling bidirectional communication with the POS data model and event infrastructure
- It is as easy as importing a JavaScript file into your web application
- Your plug-in can be presented as 'embedded' in the POS UI, or as a full-screen application

6 How to use the REST API

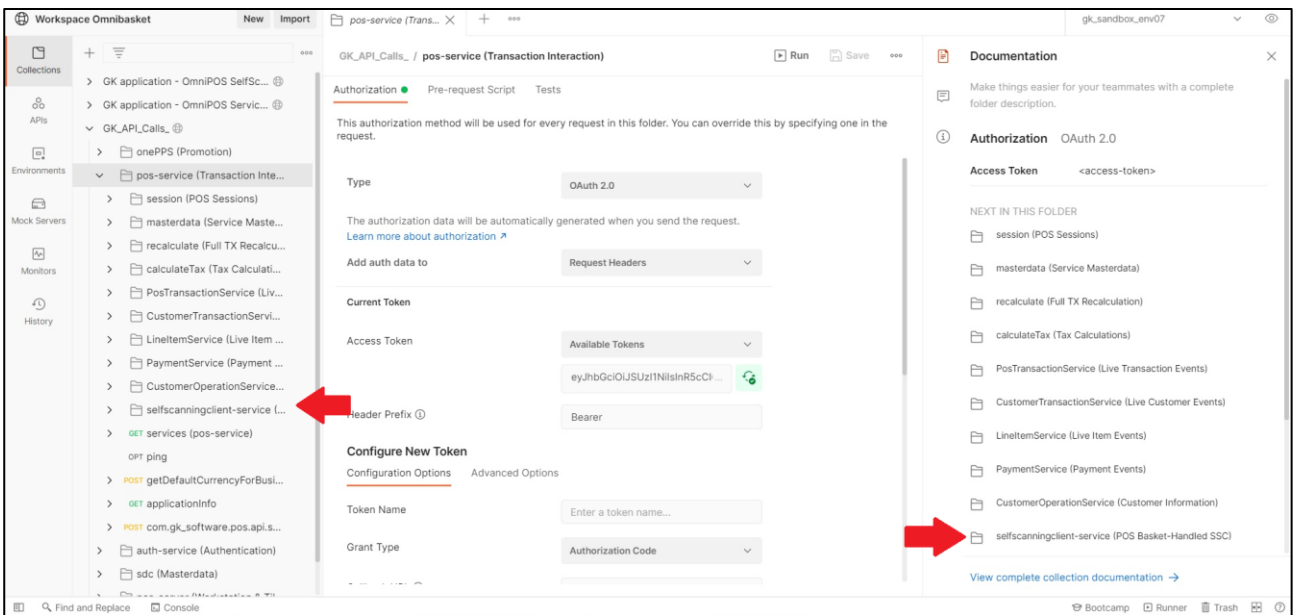
6.1 Complete Documentation

You can find a complete auto generated confluence page about the GK-REST-API right here: GK-REST-API (Please note, that the latest versions of each section is at the bottom of the drop down list)

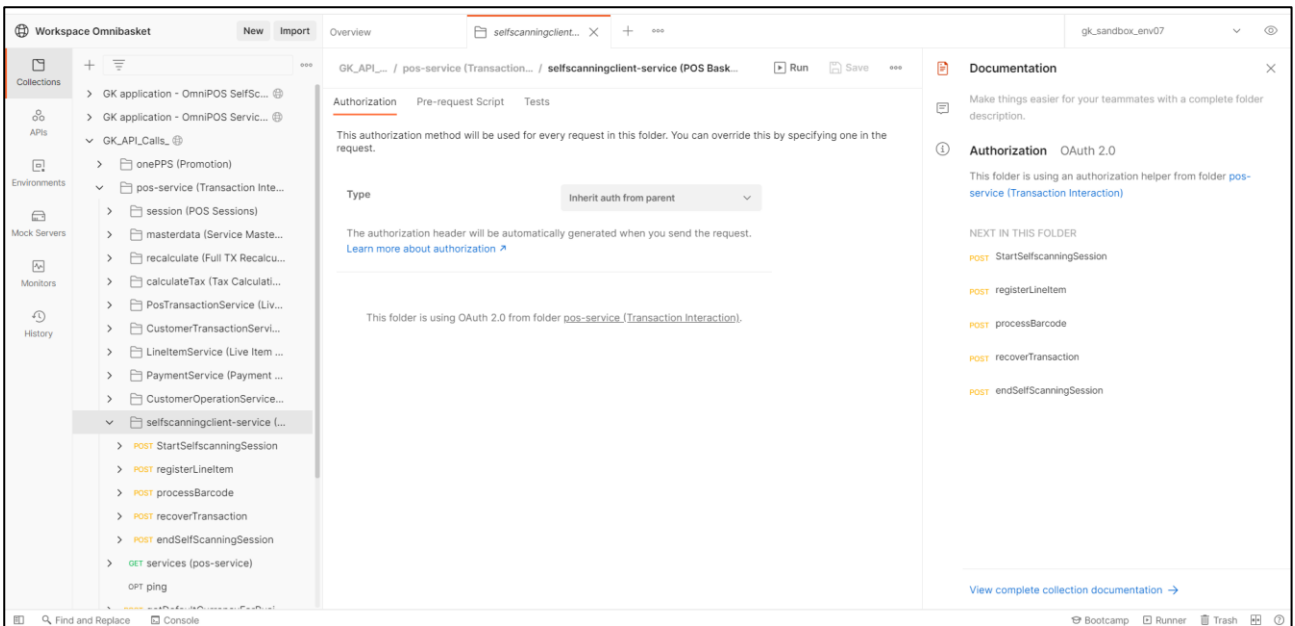
One of the most important parts of the GK_API_Calls collection, is the *pos-service (Transaction Interaction)* with the section *selfscanningclient-service (POS Basket-Handled SSC)*. To find this particular folder you will have to **click on the GK_API_Calls** Collection, so the drop down list can open. There you will see six folders, the one we need is pos-service (Transaction Interaction), **click on it**.



In there you will find another list of folders. We need the *selfscanningclient-service (POS Basket-Handled SSC)*, **click on it**. You can also see the folder on the bottom right side where the list of the following folders is. Both lead to the same folder we need.



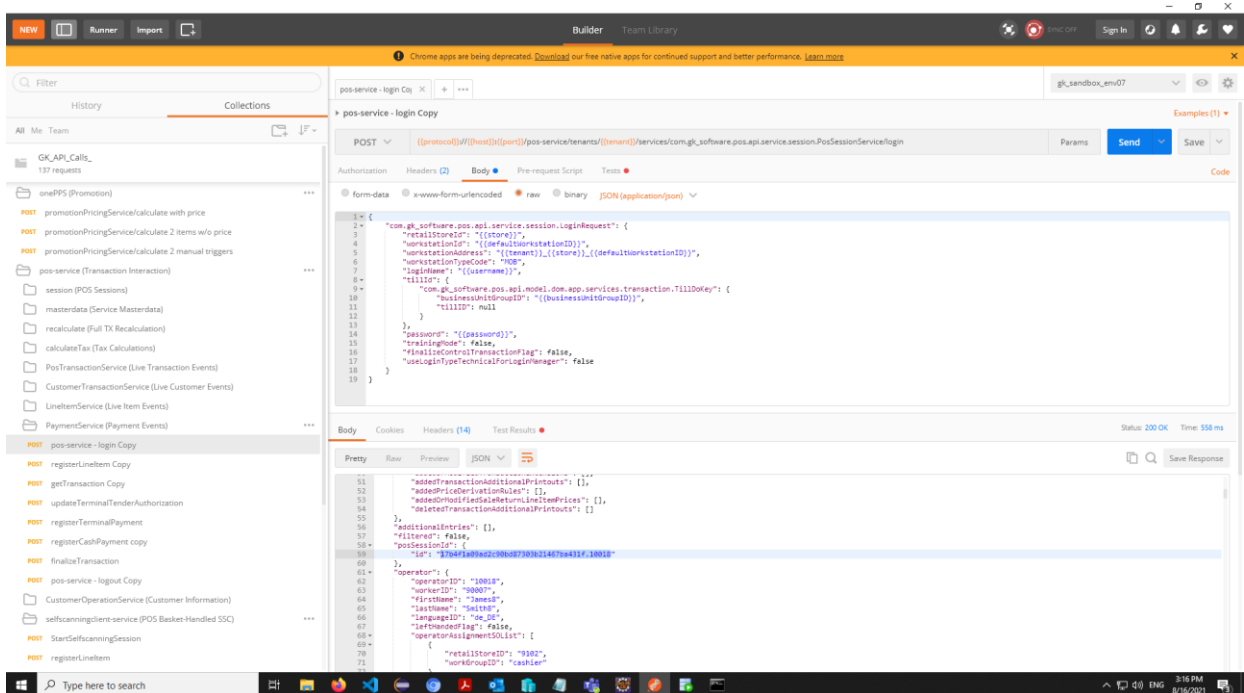
In the folder you will find a list of five POST Requests. All together they can handle the complete selfscanningclient-service.



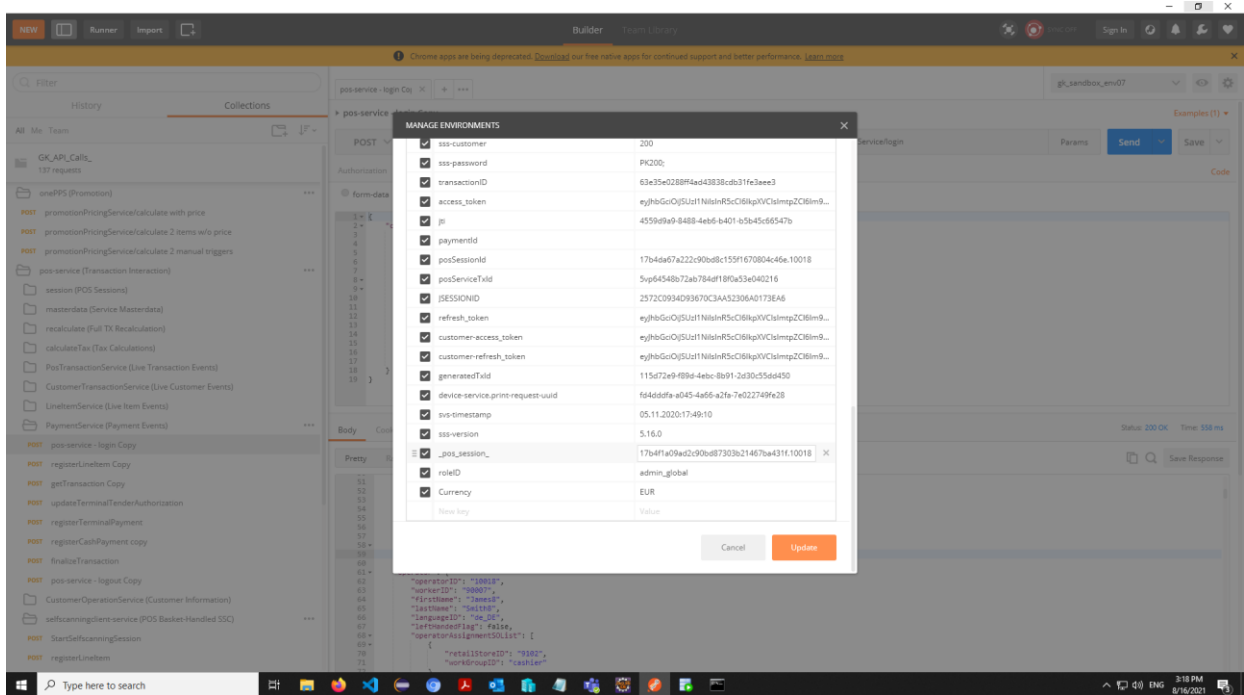
6.2 Example Request

Here we are going to complete a full use case trying to:
 login → register a line Item → get info about the current transaction → pay cash → finalize the transaction and logout at the end.
 for that please go to PaymentService folder in the collection and follow these steps.

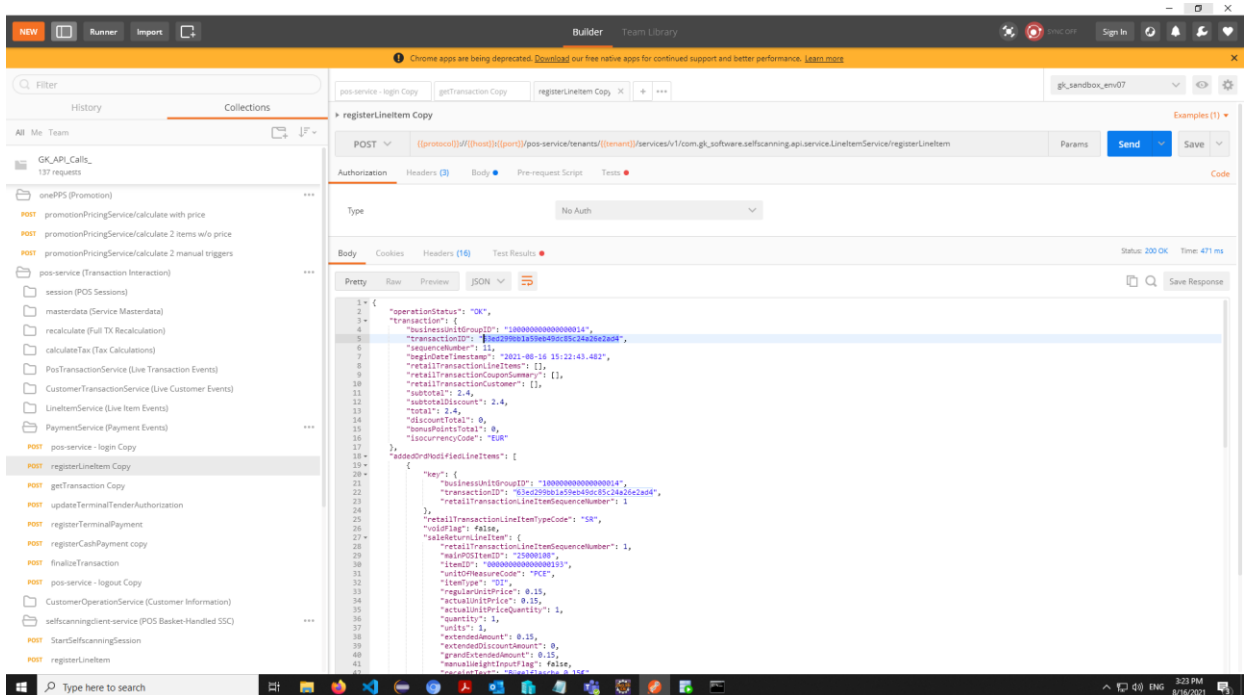
1. login and get a sessionID:



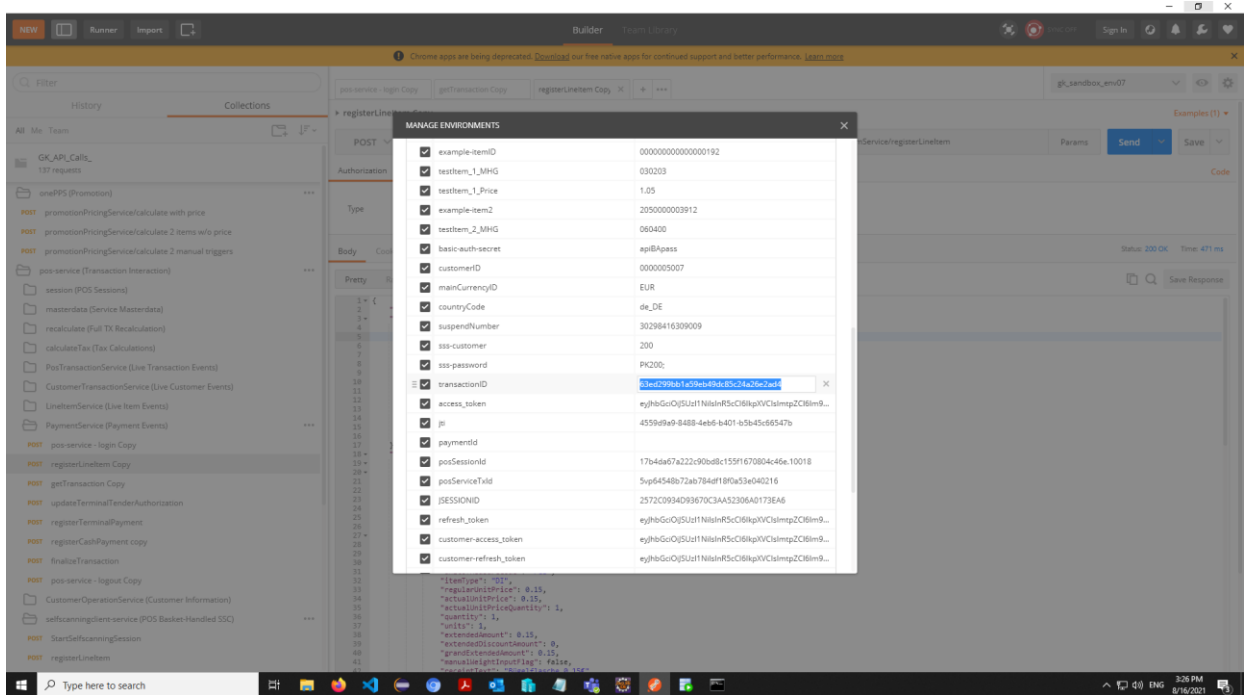
As you see, we receive the session ID, please copy it and go to your environment parameters and update the `_pos_session_` with the value you received in the response then click update



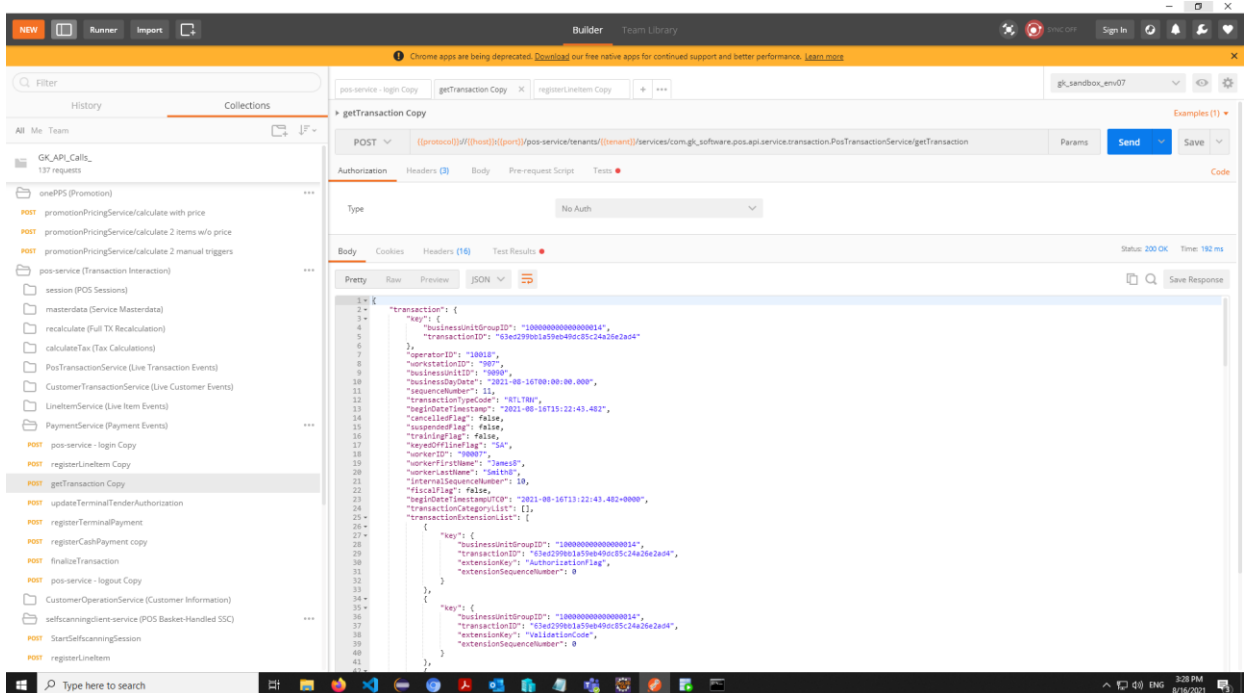
2. now you are ready to call the next endpoint (registerLineItem) to add line items to the transaction:



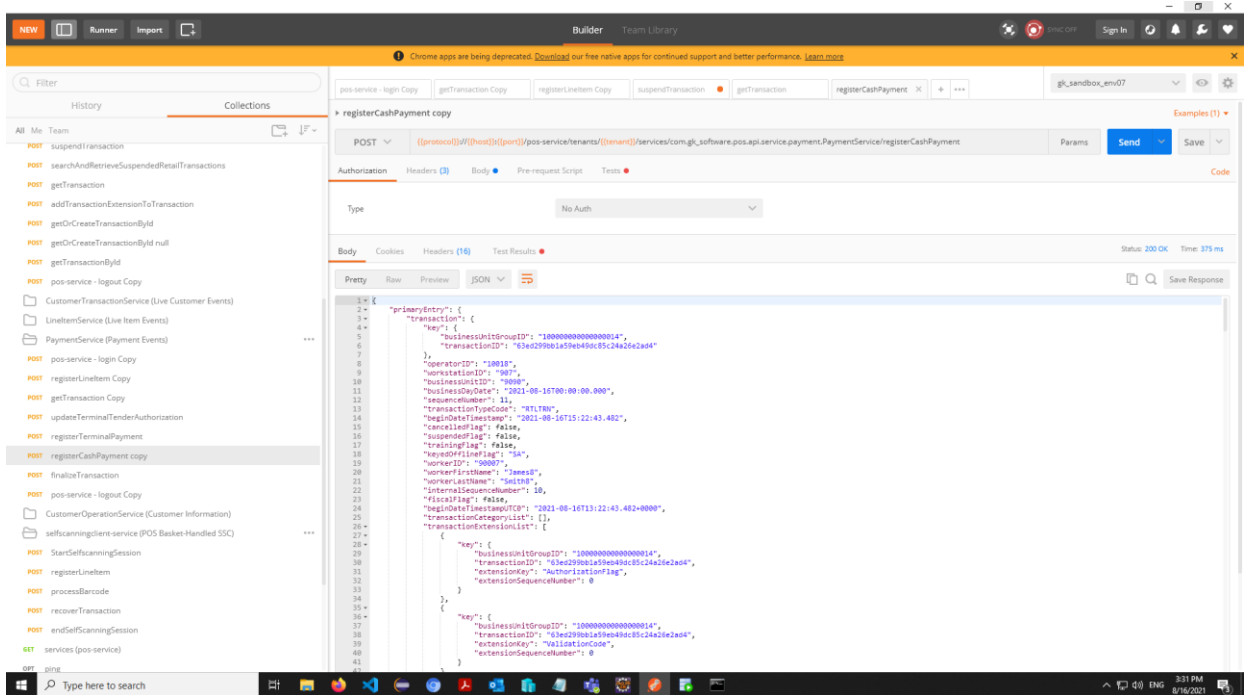
one interesting information return in the response in the transactionID which can be used to suspend or retrieve the transaction later on, so if this sound interesting to you please copy the return value of the transactionID from the response and update you environment with it :



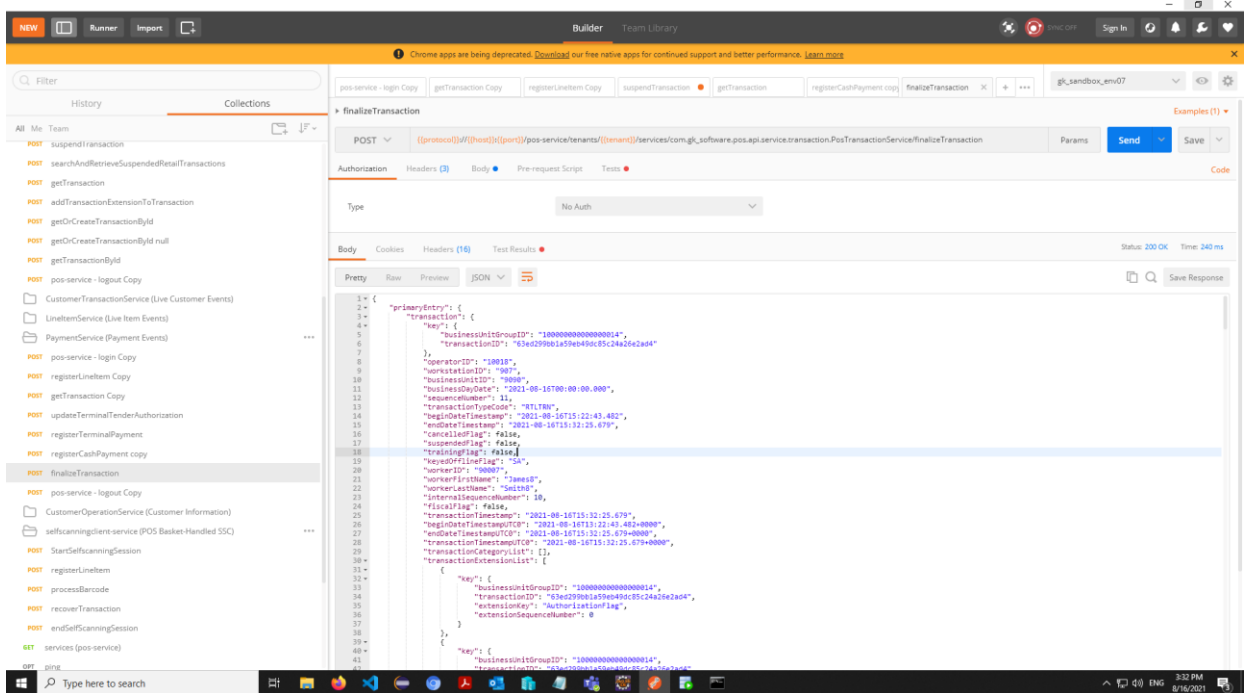
This will allow you from now on to send transaction's related request under (PosTransactionService folder) or like getTransaction endpoint in our current folder(PosTransactionService) :



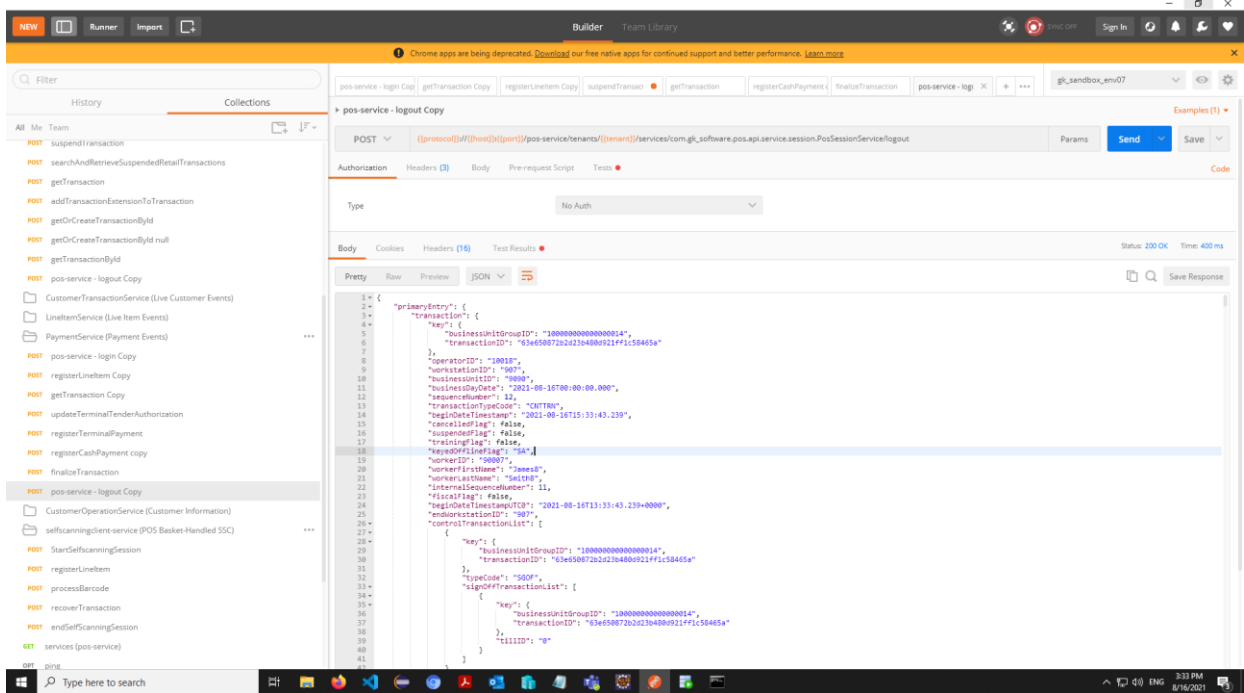
3. call the registerCashPayment :



4. Then finalize the current transaction with calling finalizeTransaction:



5. now you're finished and can call and test the pos-service logout:



CONTACT

GK Software SE
Waldstraße 7
08261 Schöneck
Germany

T +49 (0) 3 74 64 84 – 0

F +49 (0) 3 74 64 84 – 15

partner-enablement@gk-software.com

www.gk-software.com