**Transaction Pagination Solution**

**Problem Statement**

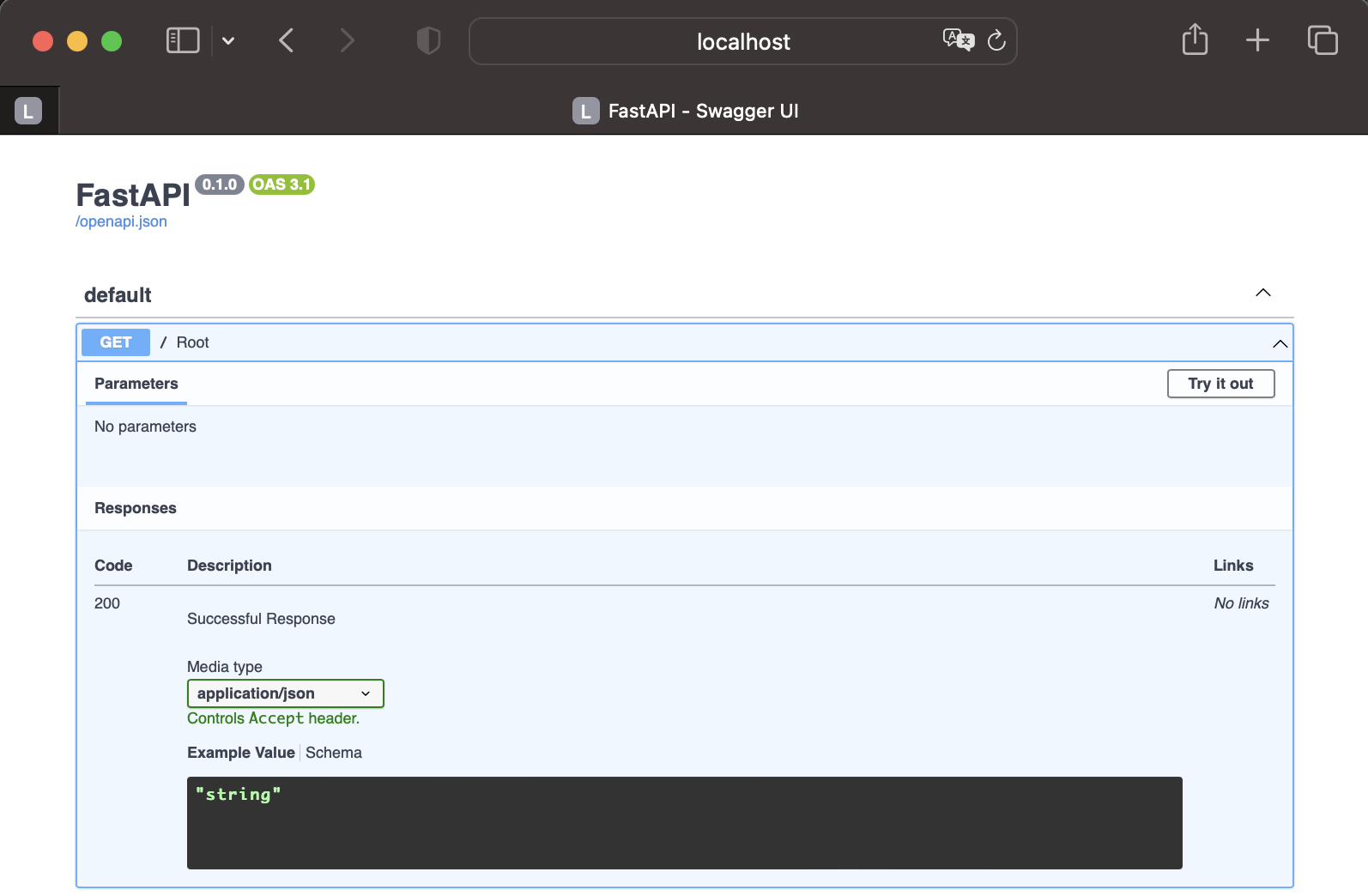
**REST API** for returning the paginated list of money account transactions created in an arbitrary calendar month for a given customer who is logged-on in the portal. For each transaction ‘page’ return the total credit and debit values with current exchange rate (from the third-party provider). The list of transactions should be consumed from a Kafka topic. The transactions cover the last ten years and are stored in Kafka with the key being the transaction ID and the value the JSON.

**Technical Approaches**

* Leveraging MongoDB system known to be Unique Key-JSON NoSql formatted by design and have built-in Kafka Topic consumption support

[*https://www.mongodb.com/docs/kafka-connector/current/source-connector/configuration-properties/kafka-topic/*](https://www.mongodb.com/docs/kafka-connector/current/source-connector/configuration-properties/kafka-topic/%20%20)

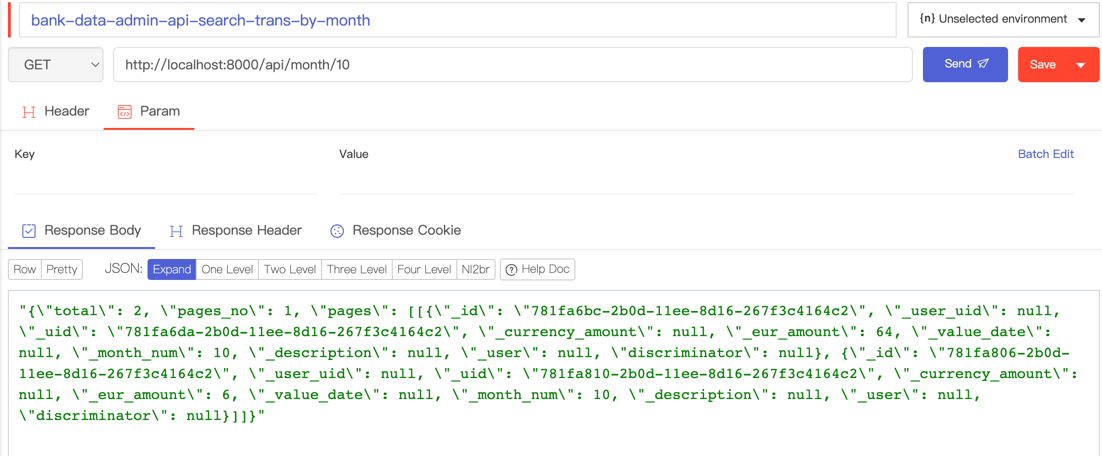
* Adding “amount\_eur” as an extra data field for Transaction data model and making transaction amount calculation within EUR as a common currency.
* Adding “debit\_credit” as an extra data field with only 1 and -1 permitted value to indicate substraction/add for debit/credit transaction amount processing
* Using Swagger generated OpenAPI data model classes to create large set of dummy transactions with randomized “amount\_eur”, transaction incurring months and debit/credit choices.
* Simplifying calendar month parameter as “month\_num” aka. 1 for Jan., 2 for Feb., etc
* Introducing 3rd party api.freecurrency.com as external API services to query for real time currency exchange rates
* Utilizing Node.js Axios asynchronous handling mechanism to address external 3rd party currency rate providers’ call response latency issues
* Leveraging FastAPI (a Python Rest solution) with built-in support for Swagger/OpenAPI documentation



* Leveraging Python’s ease-of-use Json data object, pagination data collection and float point computation processing functions

**Major Tools and Components**

**Restful API Mock**



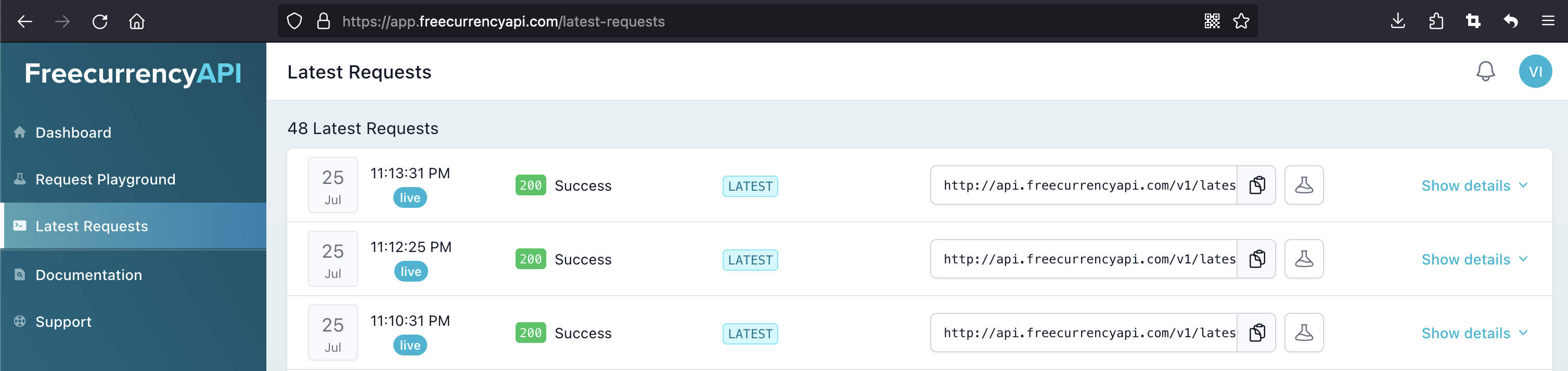
**Swagger OpenAPI Data Modeling**

SwaggerHulx 
Bank Transaction 
Info 
Tags 
Servers 
Q Search 
admins 
I.oov 
60 
61 
62 
63 
64 
65 
66 
67 
68 
69 
70 
71 
72 
73 
74 
75 
76 
78 
79 
80 
81 
82 
83 
85 
86 
87 
88 
Last Saved: 
VIRAREE_I has 9 days left in Enterprise Trial 
O 
PACTFLOW 
Upgrade 
SAVE 
searches transaction 
summary : 
operationld: searchTransaction 
description: I 
By passing in the appropriate options, 
for 
available transaction in the system 
pa rameters : 
- in: 
query 
searchString 
name : 
description: 
pass an optional search 
looking up transaction dataset 
you 
can 
search 
string 
POST 
/api 
developers 
GET 
/api 
Schemas 
SCHEMA User 
SCHEMA Transaction 
SCHEMA Str 
required: false 
schema : 
string 
type : 
in: 
query 
skip 
name : 
description: number 
schema : 
type: integer 
format : 
int32 
minimum: 
In: 
query 
limit 
name : 
maximum 
description: 
schema : 
type: integer 
format : 
int32 
minimum: 
maximum: 50 
responses : 
of 
records 
to 
skip 
for 
pagination 
number 
of 
records 
for 
to 
return 
200 
400 
search results matching criteria 
Media type 
application/json 
Controls Accept header. 
Example Value Schema 
"id": 
o, 
"user_uld" " 
• : P-Ø123456789" , 
"uid"•. "89d30179-abcd-465b-09ee-e2d5f60f 
Eld46" , 
"currency_amount": "CHF 75" , 
" eur_amount " : 
77.78, 
"value_date 
"ø1-1ø-202Ø", 
"month_num" : 
1, 
"description": "Online payment CHF" , 
"User" 
"id": 
o, 
"uid . P-ø123456789" , 
"full_name": "Peter Russells" , 
"home_address". 
"123 Sunset Avenue" , 
"phone": "408-867-5309" 
bad input parameter 
Export v 
No links 
No links 
'200' : 
1 5 am 
- Jul 23, 2023 
o 
VALID 

**MongoDB**

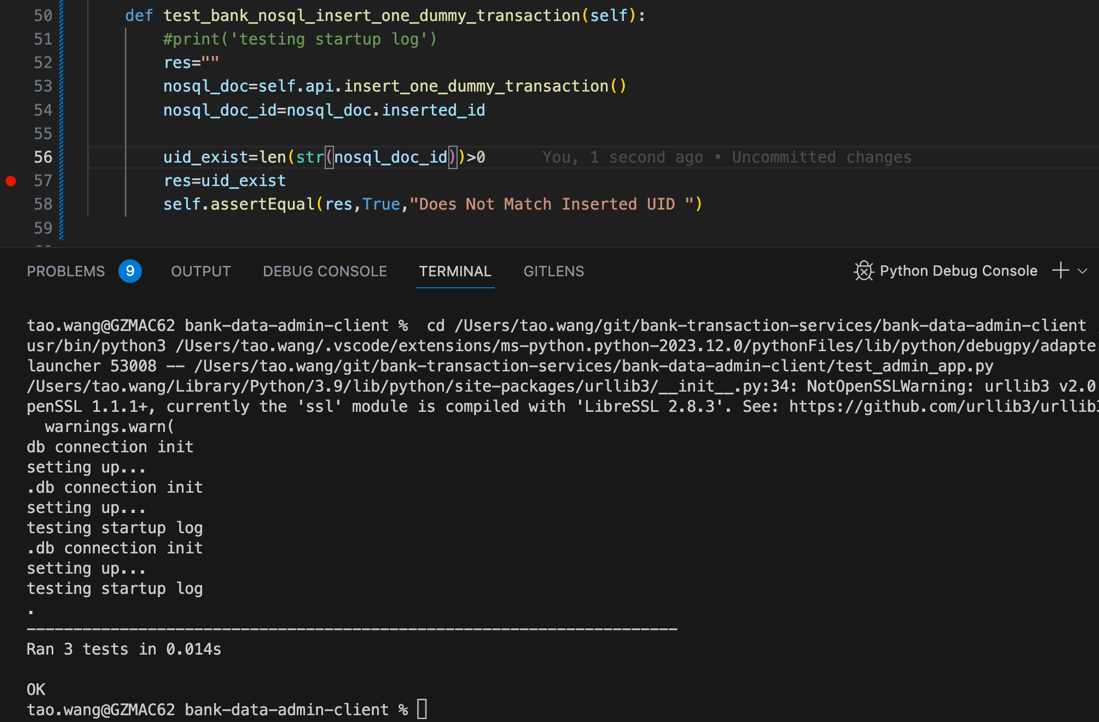
MONGODB 
v CONNECTIONS 
> % clusterO.d6b9gpg.mongo... 
> % clusterO.th7somn.mongod... 
v localhost:2017 connected 
admin 
> 
bank_nosql 
transactions 
Documents 3 
"64bd690ea452a3fc45... 
"64bd692aa452a3fc45... 
Document 3 
{ } bank_nosql.transactions:null.json X 
{} bank_nosql.transactions:null.json > 
1 
2 
3 
4 
5 
6 
7 
8 
9 
10 
11 
12 
"_id": 
null, 
"_user_uid 
'l: "P-0123456789", 
"_uid": 
null, 
" _cu rrency_amount" : 
null, 
' eur amount": 
26, 
"_value_date" : 
null, 
"_month_num" : 
3, 
" _ description": 
"_user" : 
null, 
"discriminato r": 
null, 
null 

**Node.js Axios Async**



**Test Driven Development**

v admin_app 
> api 
—init—py 
api_client.py 
configuration.py 
mongo_client.py 
rest.py 
> docs 
> swagger_client 
• 
42 
43 
44 
45 
46 
47 
48 
49 
50 
51 
52 
def 
( self) : 
dummy= swagger_client.models.Transaction() 
dummy. eu r_amount=26 
dummy. month_num=3 
dummy. user_uid= ' P—Ø123456789 ' 
collection—self. bank_nosql ' , 'transactions ' ) 
res=cot lection. inse dummyJSON) 
return res 



MONGODB 
v CONNECTIONS 
v localhost:2017 
admin 
> 
v bank_nosql 
v transactions 
{ } bank_nosqI.transactions:"3f3129e2-2b09-IIee-8c94-267f3c4164c2".json X 
connected 
1 
2 
3 
4 
5 
6 
7 
8 
9 
10 
11 
12 
"_user_uxd . 
• "P-0123456789", 
" uid": "3f312b68-2bØ9-11ee-8c94-267f3c4164c2" , 
"_cu r rency_amount" : 
null, 
"_eur_amount" : 
26, 
value date": 
null, 
"_month_num": 3, 
> 
> 
Documents 4 
"64bd690ea452a3fc45... 
"64bd692aa452a3fc45... 
Document 3 
"3f3129e2-2b09-11ee-8... 
g Schema 
Indexes 
"_description" : 
null,l 
"_use r" : 
"discriminator" : 
null, 
null 

**MQ Topic Exchange**

bank-mq-rabbit mq-server.txt 
Project 
v •e bank-mq-rabbit 
—/git/bank-transaction-services/bank-mq-rabbit 
> .settings 
v com.javainuse 
v config 
RabbitMQDirectConfig.java.bak 
RabbitMQFanoutConfig.java.bak 
RabbitMQHeaderConfig.java.bak 
a 
RabbitMQTopicConfig 
1 
mg-server. txt K RabbitMQTopicConfig.java 
docker run -d l- 
-name rabbit-mq 

**Containerized Environments**

**A screenshot of a computer

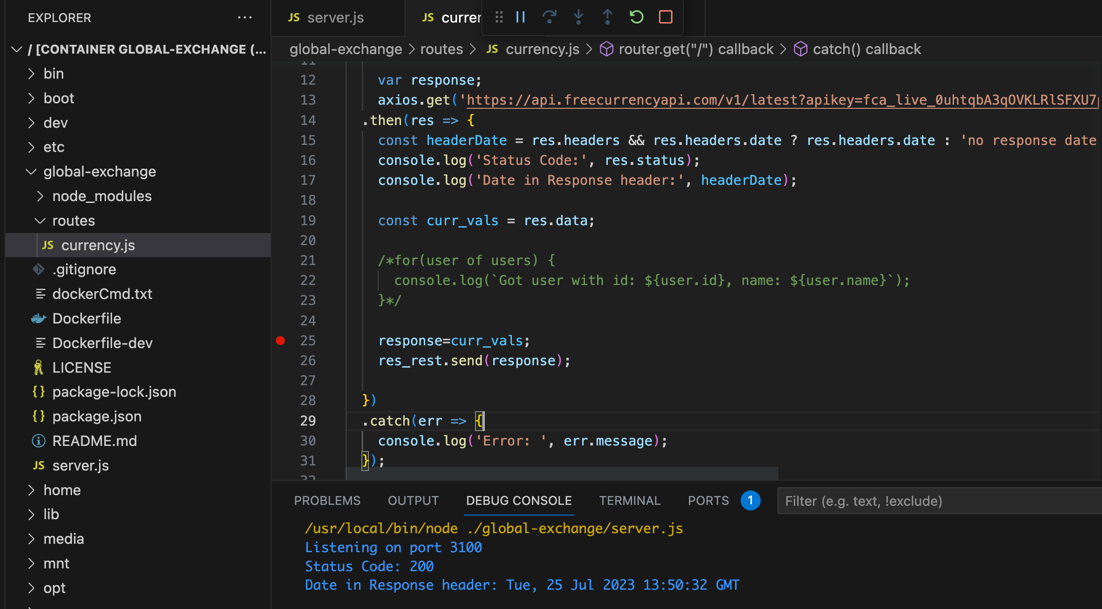
Description automatically generated**

**Implementation**

**Mock Rest API – Query Exchange Rate**



****



**Data Utility - Paginate Transaction Data Items**

15 
16 
17 
18 
19 
20 
21 
def 
paginate( items, per_page) : 
= [items [i: i+per_page] 
pages 
return { 
ten( items) , 
'total': 
'pages _ no': Len (pages), 
'pages': pages 
for i in range(ø, 
ten (items), per_page)] 

**Fetch NoSql Collection**

33 
34 
35 
36 
37 
38 
def 
fetch_collection(self, database_name, 
database=self. db [database_name] 
cot lection—database [cot lect ion_name] 
res=col lection 
return res 
col lection_name) : 

**Unit Test Fetch Collection**

MONGODB 
v CONNECTIONS 
> % clusterO.d6b9gpg.mongo... 
> % clusterO.th7somn.mongod... 
v localhost:2017 connected 
test_admin_app.py — bank-data-admin-client 
test_admin_app.py 4, M 
test_admin_app.py > TestDevelopersApi > test_bank_nosql_transaction_schema 
admin 
> 
v bank_nosql 
transactions 
> 
v PLAYGROUNDS 
No MongoDB playground files 
found in the workspace. 
Create New Playground 
v HELP AND FEEDBACK 
What's New 
Extension Documentation 
MongoDB Documentation 
Suggest a Feature 
Report a Bug 
o 
42 
43 
44 
45 
46 
47 
48 
49 
50 
51 
52 
53 
def (self) : 
print( 'testing startup log' ) 
res=" " 
collection=self. api. fetch_collect ion ( " bank_nosq l" , "transactions" ) 
res=coltection. name 
self.assertEqual( res. "transactions" , "Does Not Match 'transactions ) 
if 
name 
main 
unittest . main ( ) 
PROBLEMS 
OUTPUT 
testing startup log 
. db connection init 
setting up... 
testing startup log 
Ran 2 tests in ø.006s 
OK 
TERMINAL 
Python Debug Console + 
[1] 
tao.wang@GZMAC62 bank—data—admin—client % 

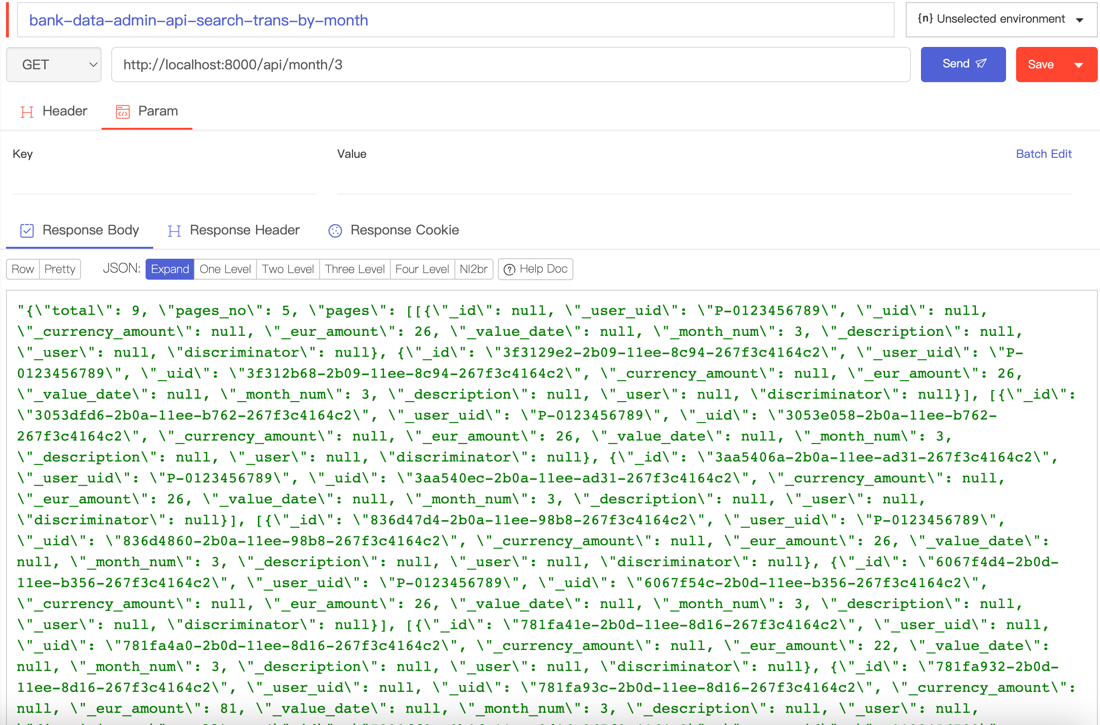
**Insert Randomized Transactions**

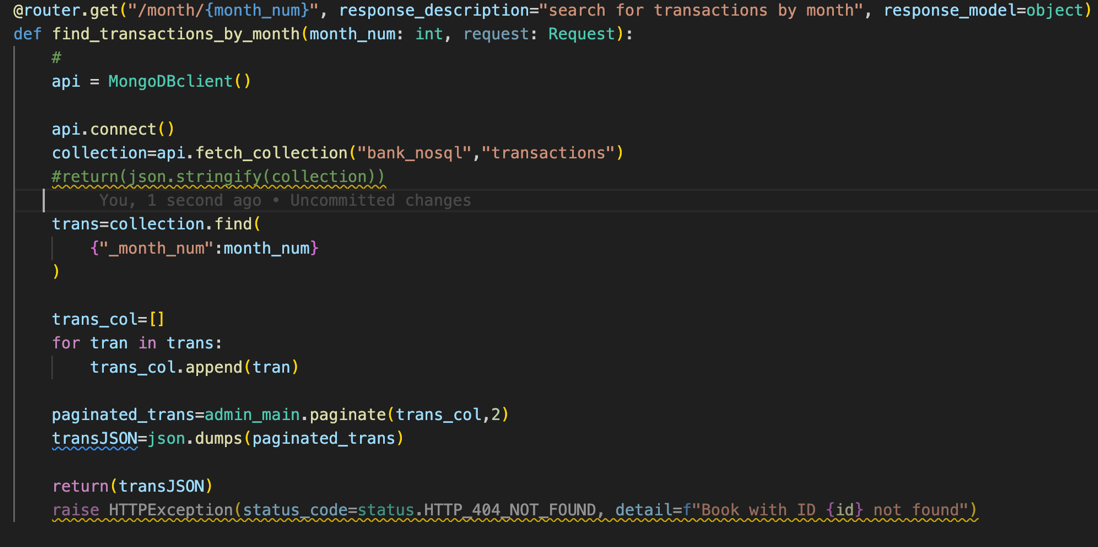
admin_app > 
57 
def 
58 
59 
60 
61 
62 
63 
64 
65 
66 
67 
68 
69 
70 
71 
72 
mongo_client.py > e MongoDBcIient > insert_many_random_transactions 
insert_many_ random_transactions (self) : 
collection—self. ' bank_nosql , 'transactions ) 
for i in range(1Ø): 
dummy= swagger_client.models. Transaction() 
dummy. id=uuid . uuidl( ) 
dummy. uid=uuid. uuidl( ) 
dummy. eu rand range( 1, 100) 
dummy. month_num=random. rand range(l, 12) 
sonable_en code r ( dummy ) 
rd _ col. append (dummyJSON) 
You, 2 minutes ago 
Uncommitted changes 
collection. insert_many( rd_col) 
res="success" 
return res 

**Unit Test Insert Many Random Transactions**

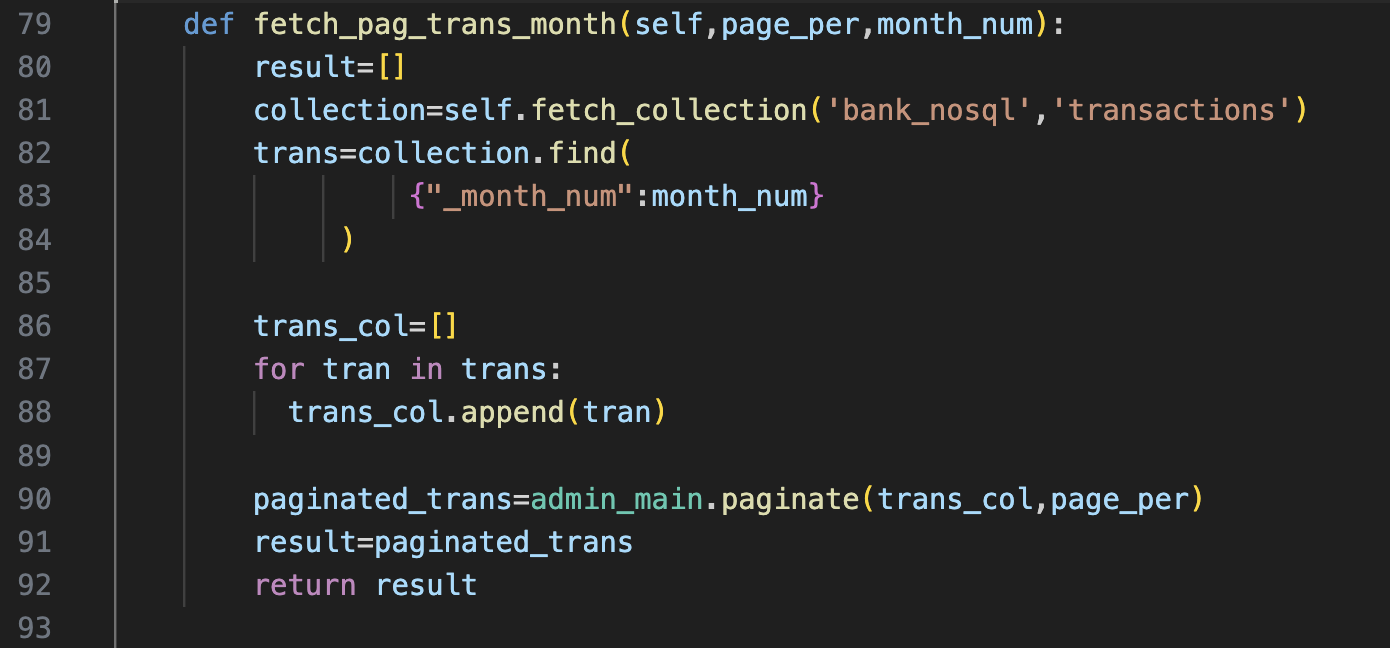
60 
61 
62 
63 
64 
65 
66 
67 
68 
def 
ransactions ( self ) : 
#print( 'testing startup log') 
res=" " 
res=se If. api. inse ransactions ( ) 
self. assertEqual( res. "success , 
" "Does Not Match 
' success ' 
if 
name 
main 
unittest. main ( ) 
TERMINAL 
GITLENS 
of Inserted Many Random") 
Python Debug Console -4- v 
PROBLEMS OUTPUT 
DEBUG CONSOLE 
db connection init 
setting up. 
.db connection init 
setting up. 
. db connection init 
setting up. 
testing startup tog 
. db connection init 
setting up. 
testing startup tog 
Ran 4 tests in 0.030s 
OK 

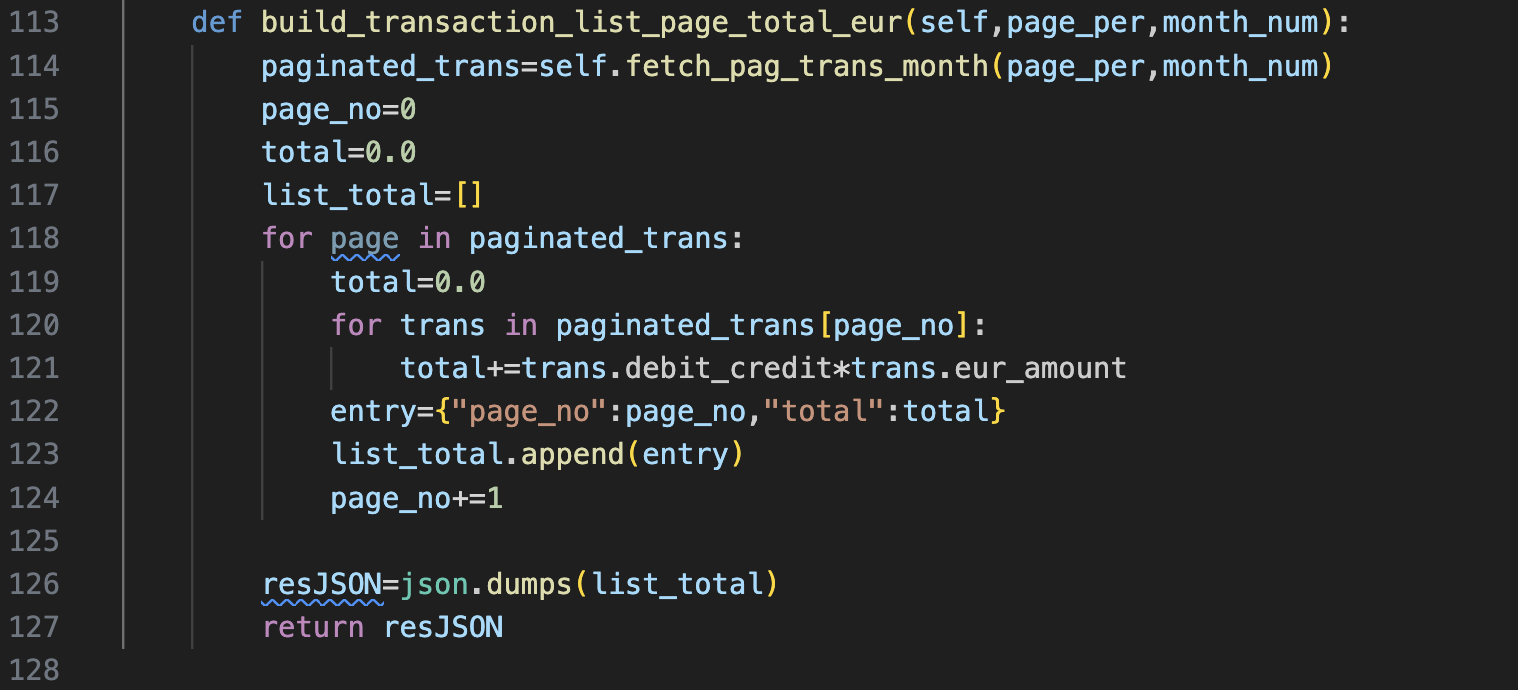
**Mock Rest API – Query for Transactions by Month**





**Calculate and List Transaction Page Totals per Page (w. EUR base)**

****

****

****