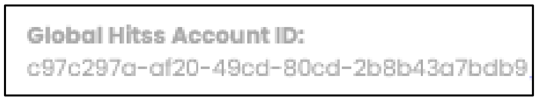
Sales Engineer Test - PayU Hub Payment Flow

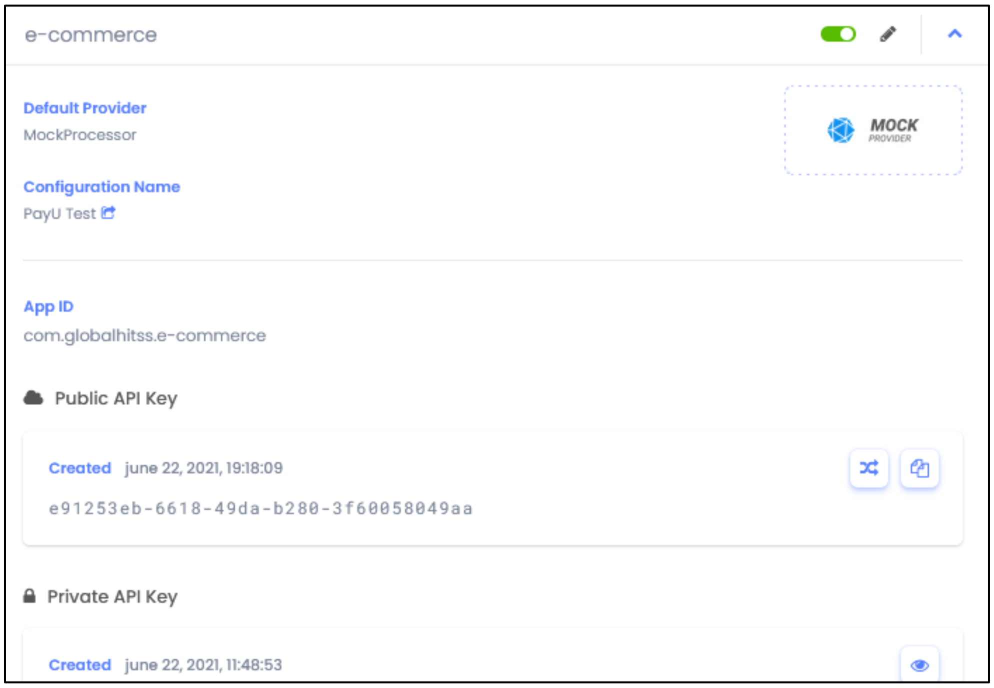
1. Sign up to the PayU Hub and set up a testing account and a Business Unit (App in the API).

I create a business unit (E-commerce) and associate a payments provider MockUp.

App Account ID:



Global Hitss Account ID:
c97c297a-af20-49cd-80cd-2b8b43a7bdb9



e-commerce

Default Provider
MockProcessor

Configuration Name
PayU Test

App ID
com.globalhitss.e-commerce

Public API Key

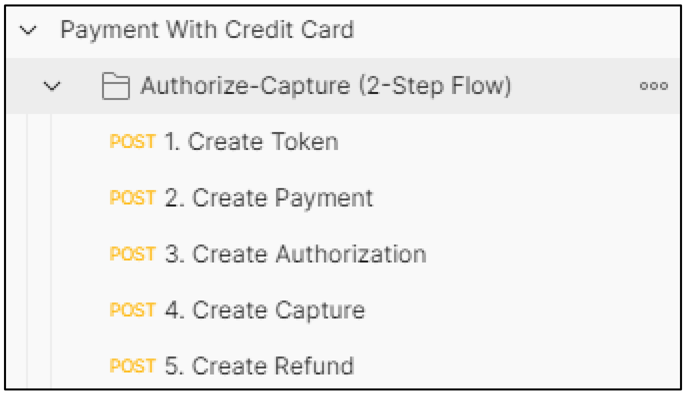
Created june 22, 2021, 19:18:09
e91253eb-6618-49da-b280-3f60058049aa

Private API Key

Created june 22, 2021, 11:48:53

2. Complete a payment flow with the following API calls (do not use the Body builder tool on this point)

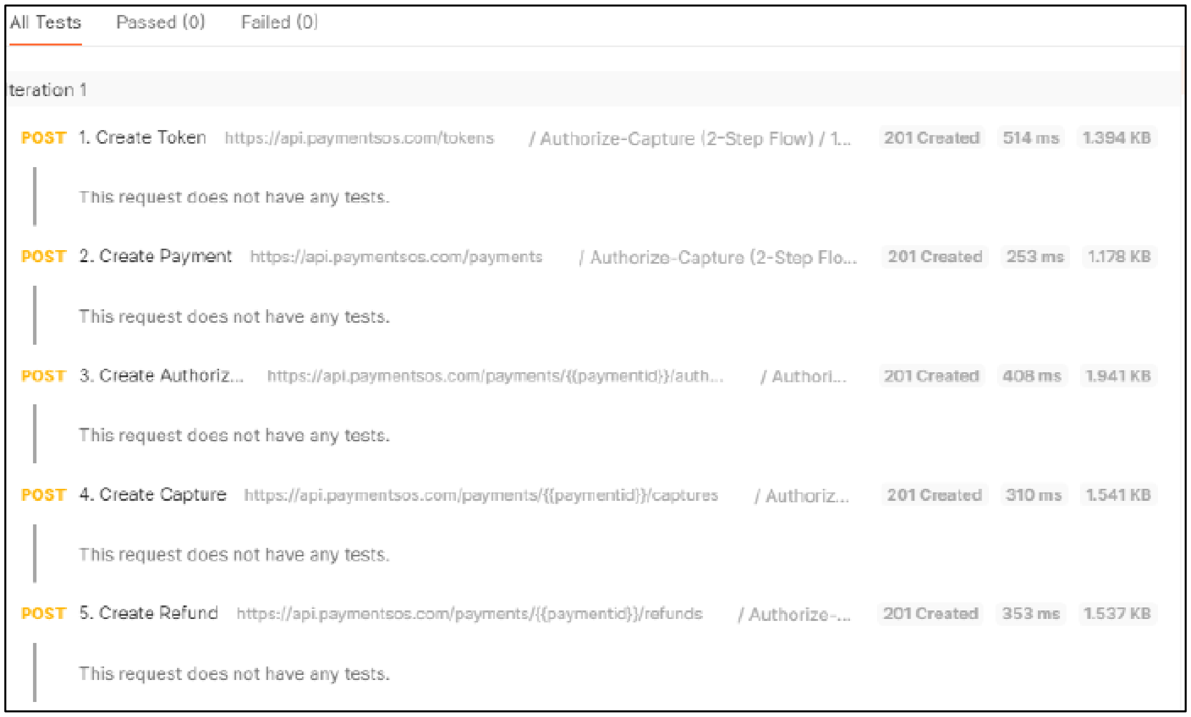
At this point i used the tool Postman that allow us to create, share, test and document APIs. This is done by allowing us to create and save simple and complex HTTP/s requests, as well as read the responses. I create a payment flow for a credit card payment with a 2 step Flow and run the request and responses.



```
Payment With Credit Card
  Authorize-Capture (2-Step Flow)
    POST 1. Create Token
    POST 2. Create Payment
    POST 3. Create Authorization
    POST 4. Create Capture
    POST 5. Create Refund
```

I launched the test on Postman.

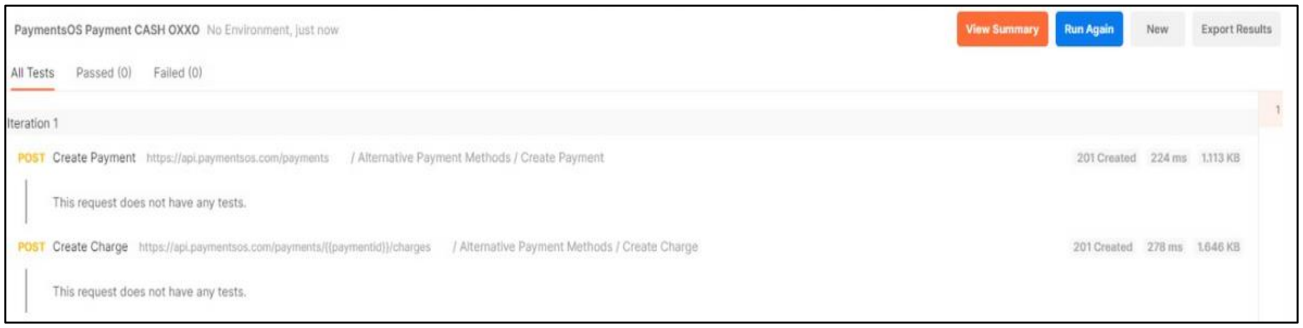
Credit Card



```
All Tests    Passed (0)    Failed (0)

teration 1
POST 1. Create Token  https://api.paymentsos.com/tokens  / Authorize-Capture (2-Step Flow) / 1...  201 Created  514 ms  1.394 KB
    This request does not have any tests.
POST 2. Create Payment  https://api.paymentsos.com/payments  / Authorize-Capture (2-Step Flo...  201 Created  253 ms  1.178 KB
    This request does not have any tests.
POST 3. Create Authoriz...  https://api.paymentsos.com/payments/{{paymentid}}/auth...  / Authori...  201 Created  408 ms  1.941 KB
    This request does not have any tests.
POST 4. Create Capture  https://api.paymentsos.com/payments/{{paymentid}}/captures  / Authoriz...  201 Created  310 ms  1.541 KB
    This request does not have any tests.
POST 5. Create Refund  https://api.paymentsos.com/payments/{{paymentid}}/refunds  / Authorize-...  201 Created  353 ms  1.537 KB
    This request does not have any tests.
```

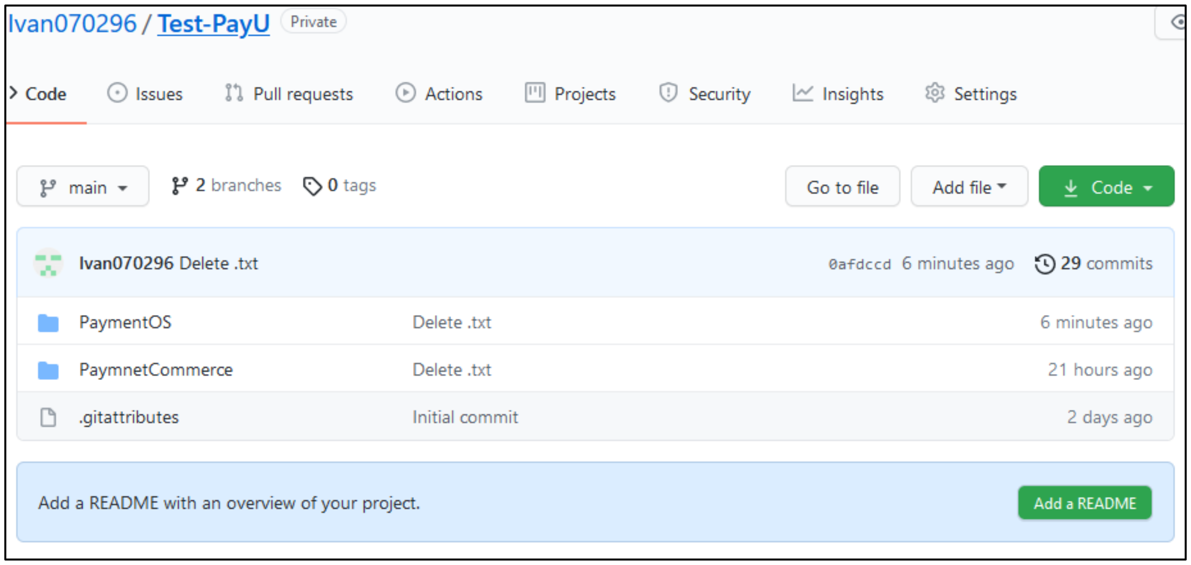
Example Cash OXXO



```
PaymentsOS Payment CASH OXXO  No Environment, just now      View Summary  Run Again  New  Export Results
All Tests    Passed (0)    Failed (0)

Iteration 1                                                                                          1
POST Create Payment  https://api.paymentsos.com/payments  / Alternative Payment Methods / Create Payment  201 Created  224 ms  1.113 KB
    This request does not have any tests.
POST Create Charge  https://api.paymentsos.com/payments/{{paymentid}}/charges  / Alternative Payment Methods / Create Charge  201 Created  278 ms  1.646 KB
    This request does not have any tests.
```

Send back your Secure fields code, all requests, and all responses. For this point, you have to use Github and share with us the Github link to review the source code.



Ivan070296 / Test-PayU Private

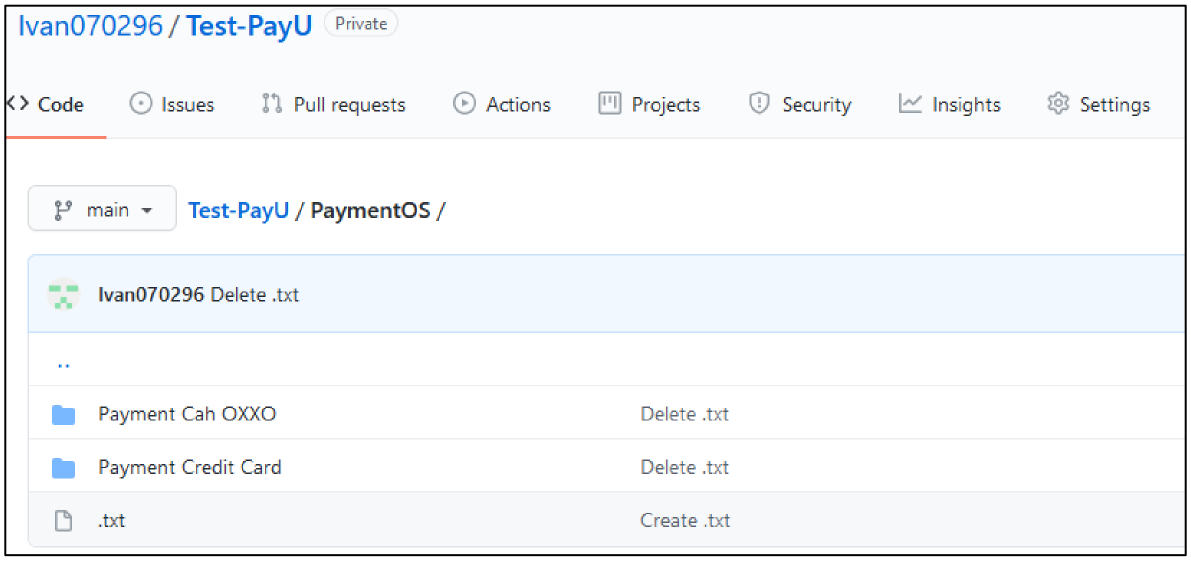
Code Issues Pull requests Actions Projects Security Insights Settings

main 2 branches 0 tags Go to file Add file Code

Ivan070296 Delete .txt 0afdccd 6 minutes ago 29 commits

PaymentOS	Delete .txt	6 minutes ago
PaymnetCommerce	Delete .txt	21 hours ago
.gitattributes	Initial commit	2 days ago

Add a README with an overview of your project. Add a README



Ivan070296 / Test-PayU Private

Code Issues Pull requests Actions Projects Security Insights Settings

main Test-PayU / PaymentOS /

Ivan070296 Delete .txt

..

Payment Cah OXXO	Delete .txt
Payment Credit Card	Delete .txt
.txt	Create .txt

I have uploaded the files with the request and responses to my public repository at Github.

https://github.com/Ivan070296/Test-PayU

PAYMENT COMMERCE:

This folder contains the files of a website that generates a token by calling APIs.

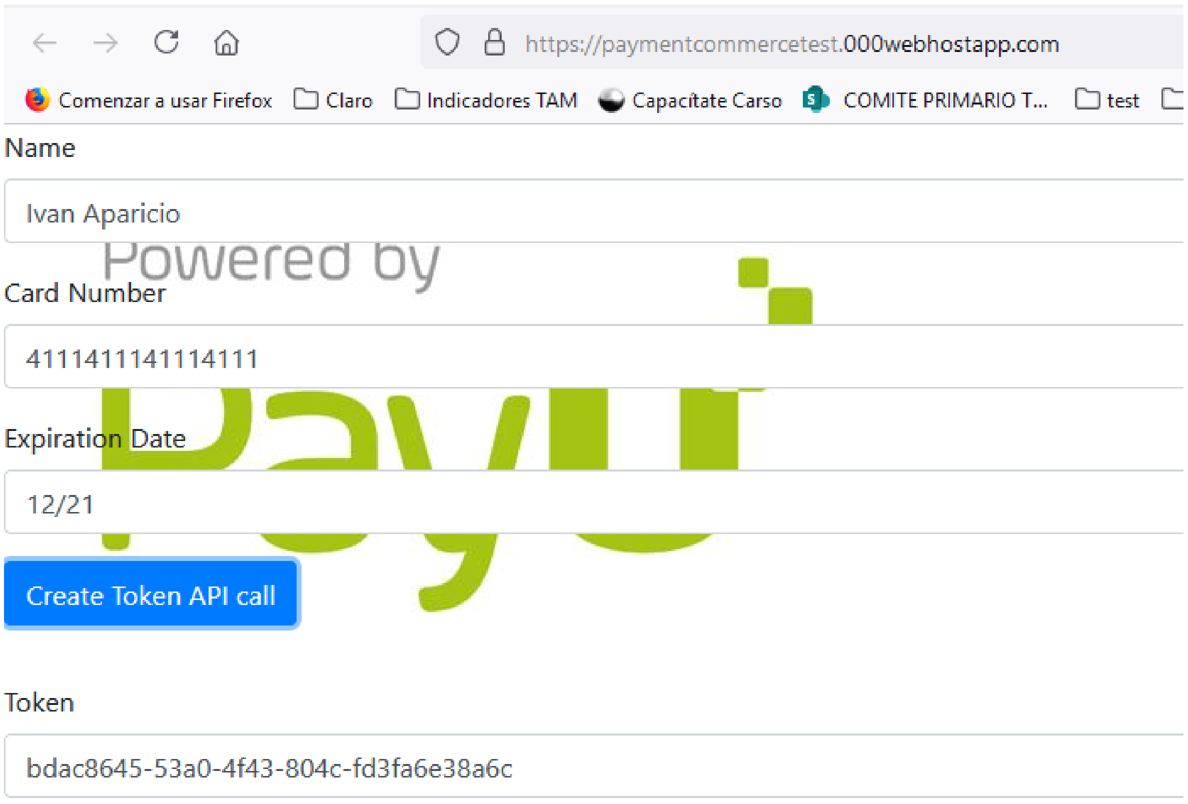
PAYMENTSOS :

Thohis folder contains the files exported from Postman with the calls to the APIs and the tests results.

3. To test the secure fields, form feature, please upload the secure fields form code in a public hosting and share the URL with us. X

I deployed a simple website that use the API calls for this, instead Secure Field Form feature.

URL site: https://paymentcommercetest.000webhostapp.com/



Name

Ivan Aparicio

Card Number

4111411141114111

Expiration Date

12/21

Token

bdac8645-53a0-4f43-804c-fd3fa6e38a6c

4. Keeping in mind your experience with this Technical Assessment, please share with us what is the definition of tokenization, authorize, capture, charge,void, and refund.

TOKENIZATION: Refers to converting payment data (confidential card details) into a token (code), in this way, the client's real data remains protected during the online transaction, preventing it from being intercepted for illegitimate purposes and travels through the platforms and payment networks to make the operation effective while the client´s data remains protected.

We have to keep in mind that token is unique and can only be used within the platform or device for which it was generated. In addition, it is irreversible, so it has no value and would not allow purchases to be made on behalf of the customer if it were intercepted.

AUTHORIZE: Process where it is verified together with the bank if the purchase can be made. it is confirmed if the card is active and has funds.

CAPTURE: At this stage the purchase is confirmed and charged to the credit card. only after making the capture the payer will be able to see the purchase reflected in his bank account.

AUTHORIZE AND CAPTURE: Operation where the two processes previously described are performed simultaneously, is the most used type of transaction. With this option, the amount of the transaction is sent to authorization and if it is approved, the capture is carried out immediately.

CHARGE: Is an one-step payment operation that combines the Authorize-Capture steps into a single transaction. Funds are transferred from your customer's account to your acquiring bank once the payment has been authorized.

VOID: Cancels an operation such as an authorization or capture, before it has been finalized. The most common procedure is to void an authorization.

REFUND: Refund allows you to cancel or undo a sale and return the charged amount to the customer.

The main different between refund and void is with void transactions, no money is ever actually transferred from the customer’s debit or credit card company to the merchant. But refunds are issued after a transaction has settled and the customer has paid for the good or service.

Some merchants and credit card processing systems may actually settle transactions immediately. When a transaction settles immediately, the seller must issue a refund rather than voiding the transaction.

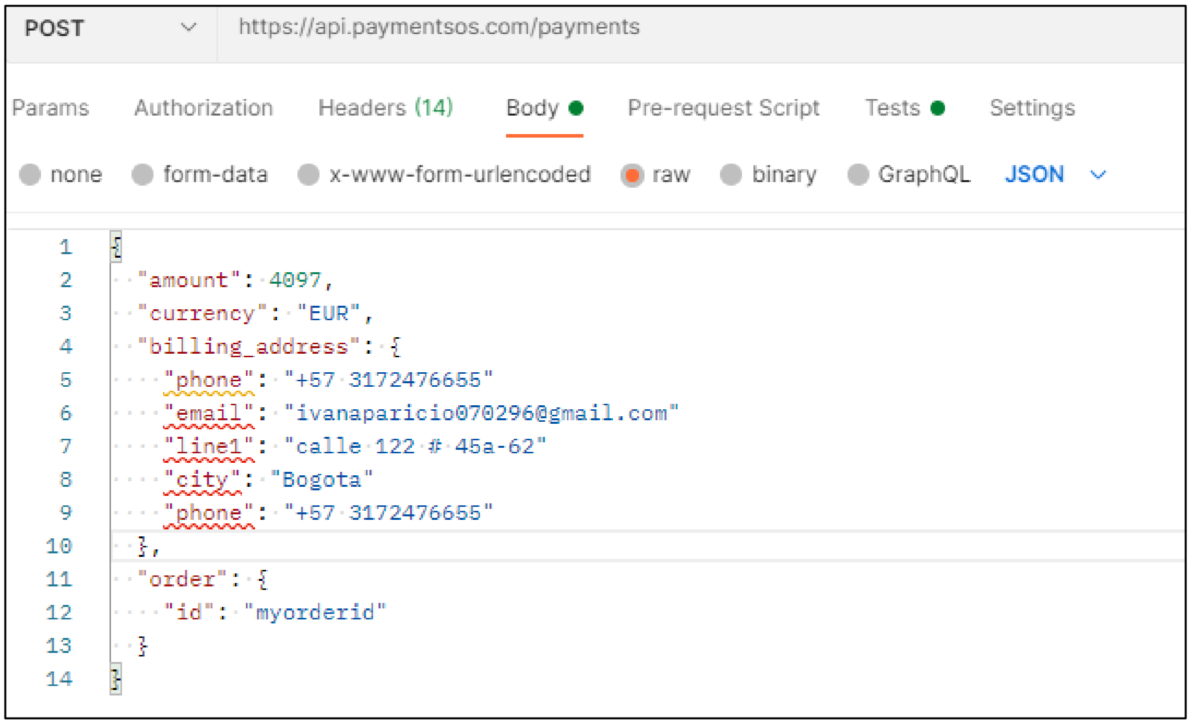
Unlike void transactions, refunds can take a much longer pass through to a customer's account. Some refunds take as little as 48 hours to reflect on a customer's account, while others can take a few days.

5. And finally, share with us your merchant onboarding, integration and documentation experience notes on a file on the same github resource.

This document will be uploaded at https://github.com/Ivan070296/Test-PayU plus points

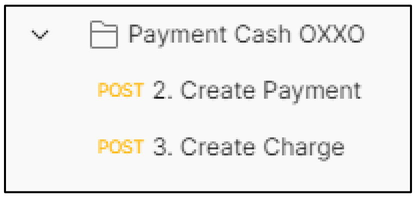
6. Add the billing information (email, line1, city, country, phone) to the secure fields form.

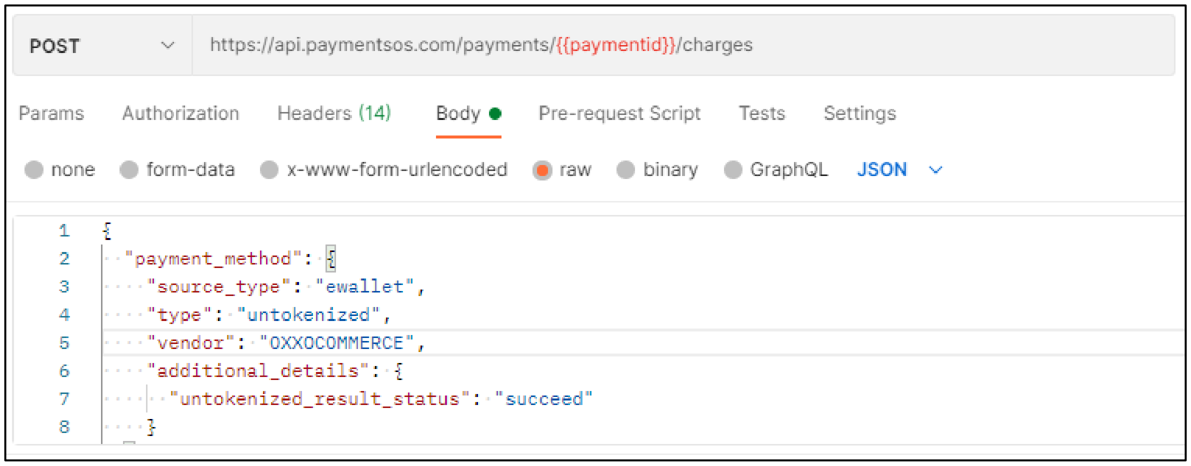
I have added the billing data to the Creation Payment step with credit card:



```json
{
  "amount": 4097,
  "currency": "EUR",
  "billing_address": {
    "phone": "+57 3172476655",
    "email": "ivanaparicio070296@gmail.com",
    "line1": "calle 122 # 45a-62",
    "city": "Bogota",
    "phone": "+57 3172476655"
  },
  "order": {
    "id": "myorderid"
  }
}
```

7. Create a postman collection and a sequence diagram (two actors, merchant and payu) explaining the flows for the following scenario: payu has a merchant in mexico that wants to integrate credit cards to process without cvv(security code) and cash payments(oxxo).

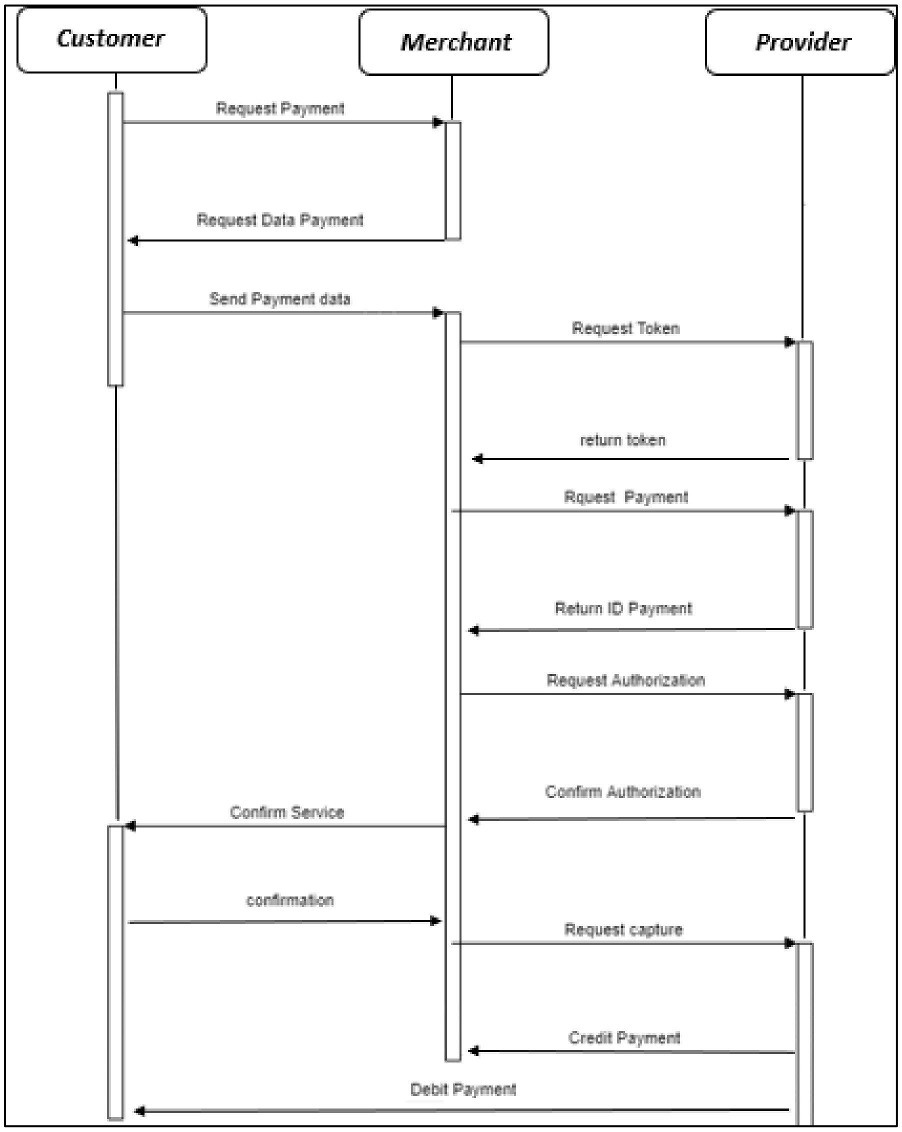




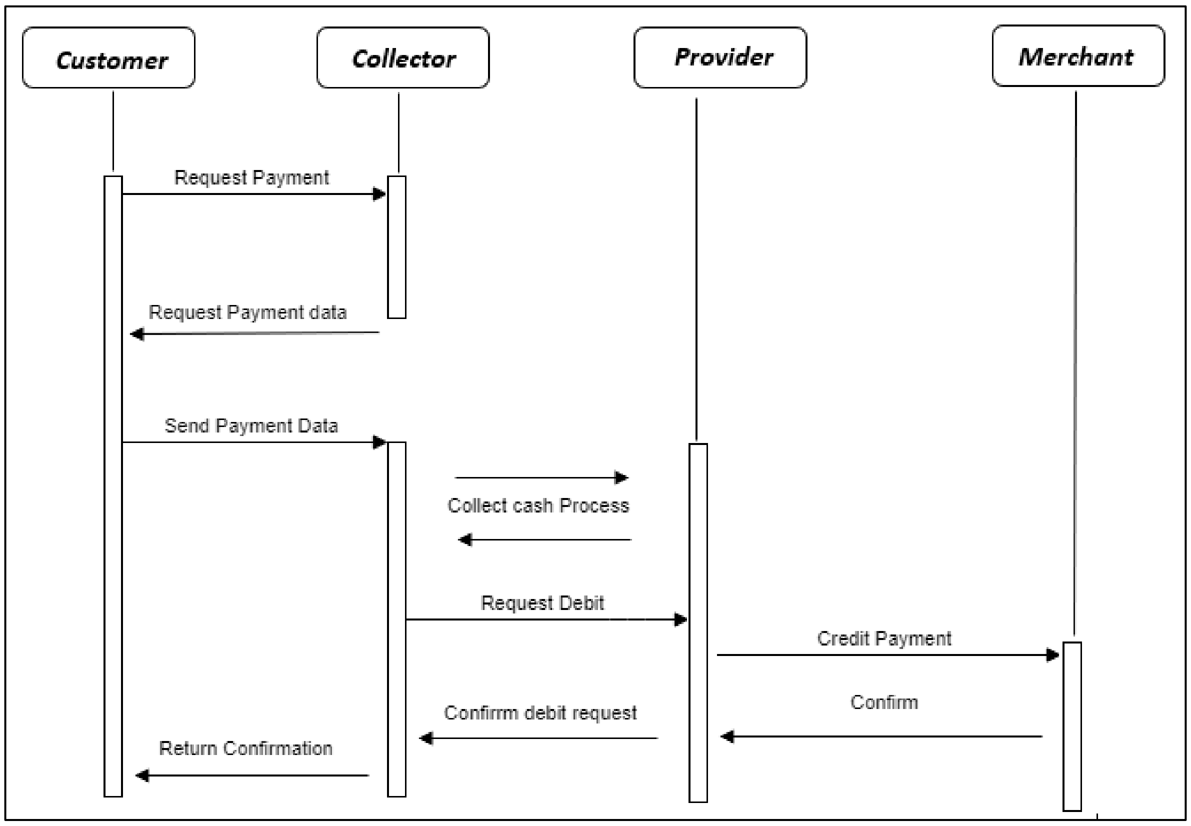
https://api.paymentsos.com/payments/{{paymentid}}/charges

```
{
  "payment_method": {
    "source_type": "ewallet",
    "type": "untokenized",
    "vendor": "OXXOCOMMERCE",
    "additional_details": {
      "untokenized_result_status": "succeed"
    }
```

CREDITCARD PAYMENT 2 STEP FLOW SEQUENCE DIAGRAM



CASH PAYMENTS SEQUENCE DIAGRAM (OXXO COMMERCE)


Customer
Collector
Provider
Merchant
Request Payment
Request Payment data
Send Payment Data
Collect cash Process
Request Debit
Credit Payment
Confirm
Confirm debit request
Return Confirmation