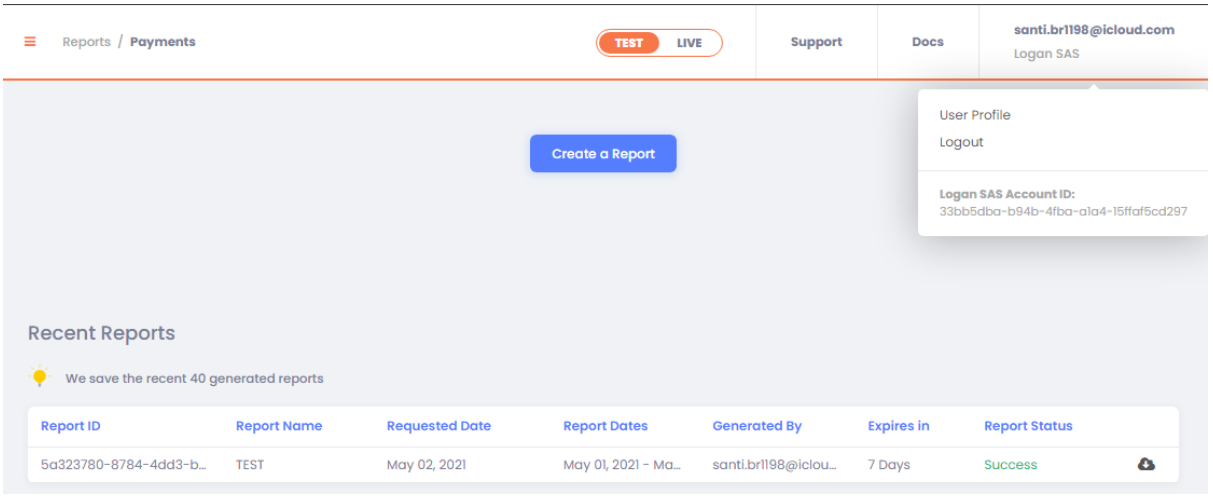
Sales Engineer Test - PayU Hub Payment Flow Santi.br1198@icloud.com
David Bohorquez

***README:* (Solution)**

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



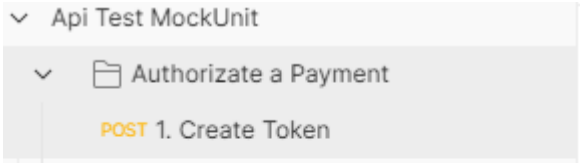
Control panel: https://control.paymentsos.com/signup .
Please, share with us the account_id created.

The account_id created :

Logan SAS Account ID: // Name only for the exercise
33bb5dba-b94b-4fba-a1a4-15ffaf5cd297

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

1. Create Token



The client is Mr. Santi Logan from FL, USA his Method of Payment is a Credit Card with this data :

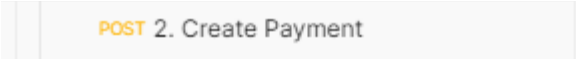
```
"credit_card_cvv": "098",
"card_number": "4000620000000007",
"holder_name": "Santi Logan",
"expiration_date": "03/30"
```

after the API CALLs we have the Card Tokenized :

```
"token": "df5454de-ea88-46a5-b5d2-5c7c03c3e642"
```

Doc used : https://developers.paymentsos.com/docs/apis/payments/1.3.0/#tag/Tokens

2. Create Payment

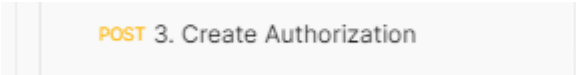


The client need to pay 2000 USD and the API allow a Payment ID :

"id": "7f34bf72-ce10-4782-8cad-bc589457c167"

Doc used : https://developers.paymentsos.com/docs/apis/payments/1.3.0/#tag/Payments

3. Create Authorize



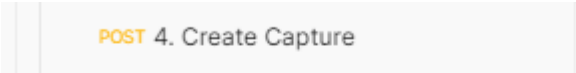
For the Authorization of payment , send to API the Tokenized Data of Credit Card

"credit_card_cvv": "098"
"token": "df5454de-ea88-46a5-b5d2-5c7c03c3e642"

The response of the API is the Success of the Transaction.

Doc used : https://developers.paymentsos.com/docs/apis/payments/1.3.0/#tag/Authorizations

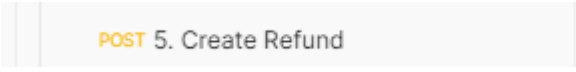
4. Create Capture



To Capture is just Put the Value to capture and the Key is the PaymentID

Doc used : https://developers.paymentsos.com/docs/apis/payments/1.3.0/#operation/create-a-capture

5. Create a Refund



To Refund is just Put the Value to refund and the Key is the PaymentID

Doc used : https://developers.paymentsos.com/docs/apis/payments/1.3.0/#operation/create-a-refund

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.

Link GitHub : https://github.com/Thorhan77/PayUTest.git

Sales Engineer Test - PayU Hub Payment Flow Santi.br1198@icloud.com
David Bohorquez

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

Link GitHub : https://github.com/Thorhan77/PayUTest.git

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 : is a process where the main functionality is not to compromise the data of the card or payment method but to assign them a token. When generating the token, the information of the credit card or payment method of the clients is saved, the Payment gateway (PayU) checks that the payment information is correct and then a Token is delivered to the Merchant to either authorize, capture , refund or charge an amount.

Authorize : is a process used to verify if a credit card is active with the Token and Payment Amount, basically check if the card has funds, The transaction is not finalized until a capture transaction is submitted.

Capture : is a process that ends a previously authorized transaction. This is when the card account is debited.

Charge : Is the result of Authorize process and Capture. in this Process the amount is debited of the Card.

Void : is a process to void an authorized payment before it is captured on a Credit Card.

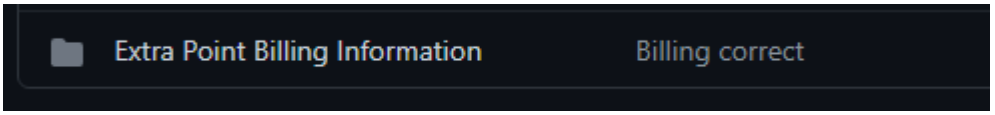
Refund : is a process to Refund an specific amount payment after the card account is debited.

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

Link GitHub : https://github.com/Thorhan77/PayUTest.git

Plus points :

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



Link GitHub : https://github.com/Thorhan77/PayUTest.git

***2.** 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).*

Sales Engineer Test - PayU Hub Payment Flow
David Bohorquez

Santi.br1198@icloud.com

Send the URL of the postman collection with folders one for Cash and the other for credit cards.

Credit cards to process without :

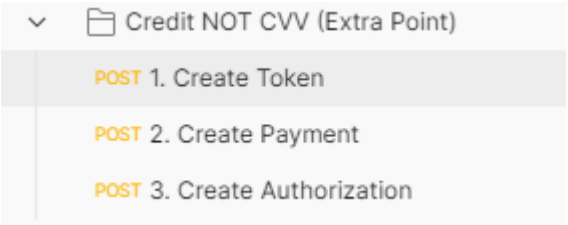
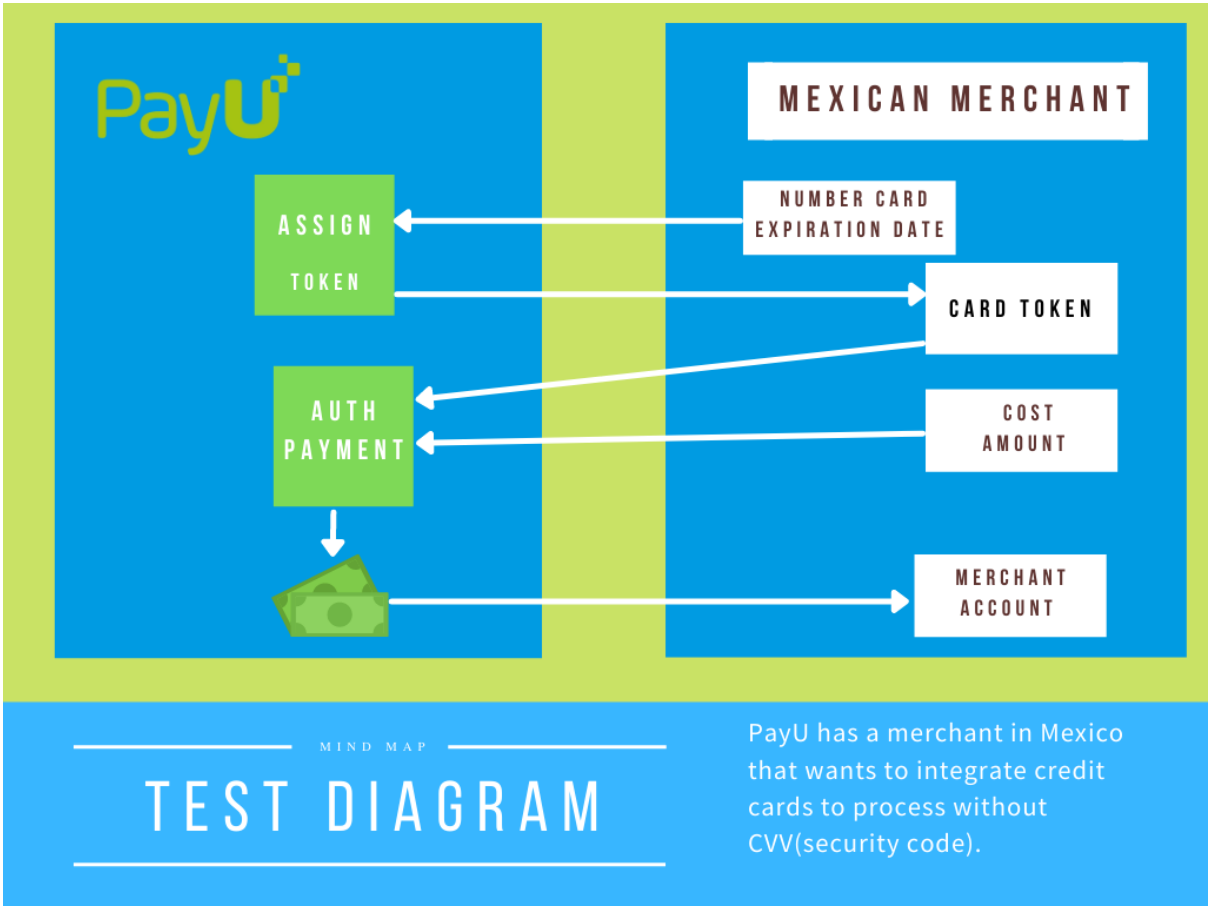
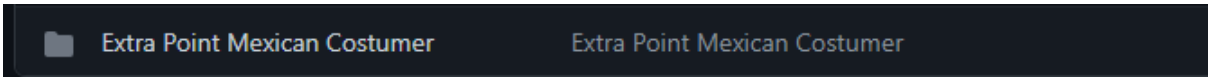


Diagram :



URL of Postman Collection : https://www.getpostman.com/collections/8558f79650106f9e299a



Link GitHub : https://github.com/Thorhan77/PayUTest.git

Sales Engineer Test - PayU Hub Payment Flow
David Bohorquez

Santi.br1198@icloud.com

Cash payments :

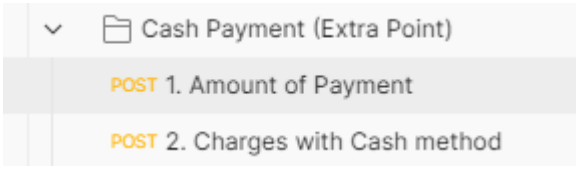
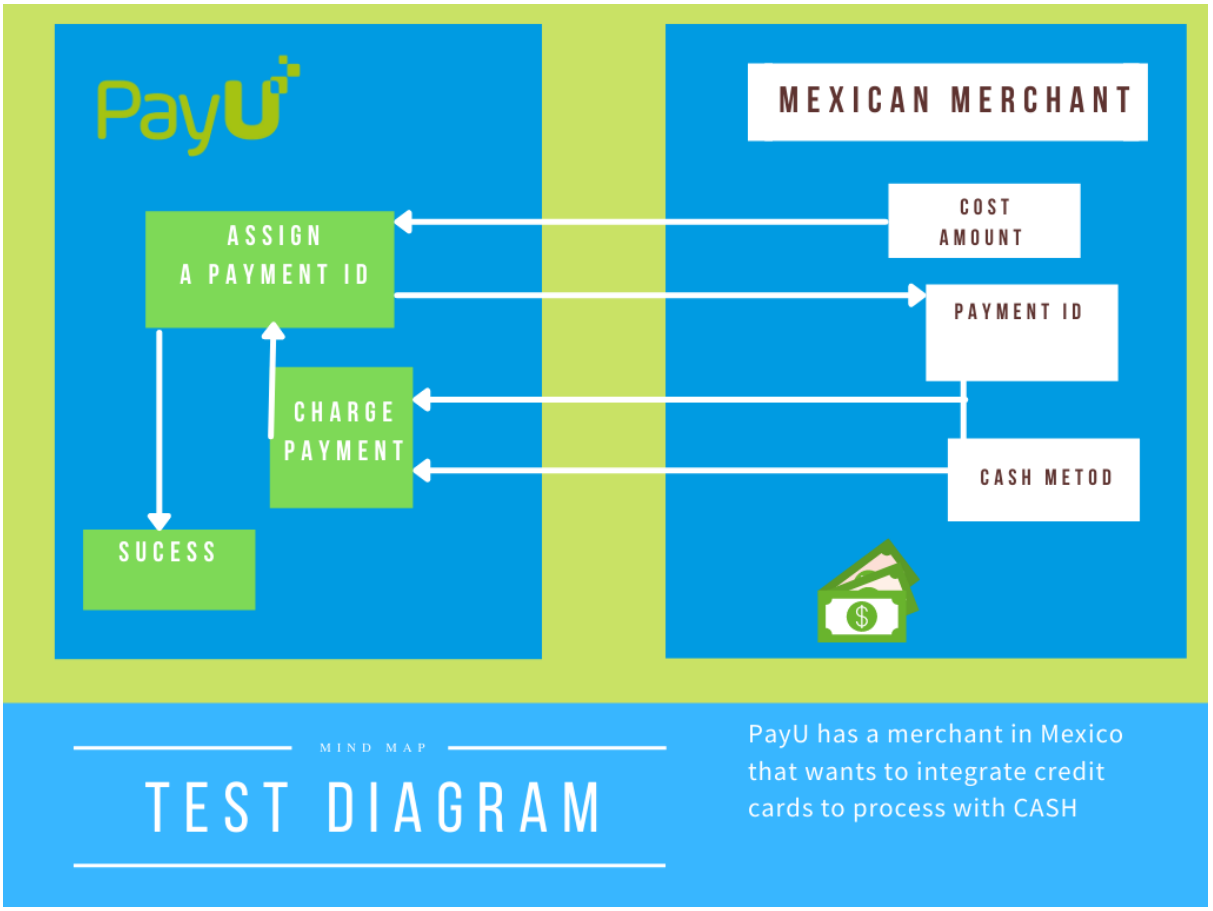
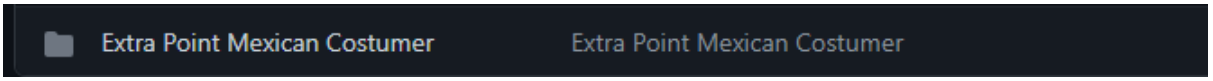


Diagram :



URL of Postman Collection : https://www.getpostman.com/collections/8558f79650106f9e299a



Link GitHub : https://github.com/Thorhan77/PayUTest.git