Maya Direct Biller- Pre-Integration Testing Tool
User Guide

Table of Contents

Pre-requisites

- Postman App
  - https://www.postman.com/downloads/
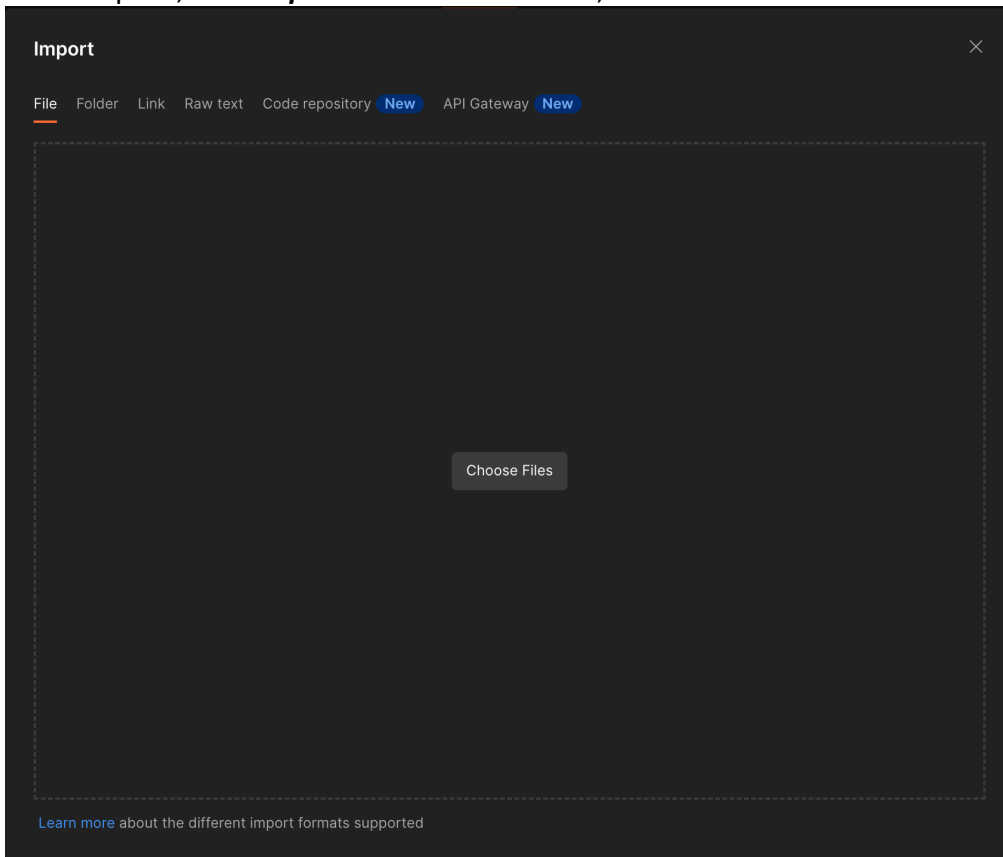- Maya Bills Pay Direct Biller Postman Collection
  - https://cdn.paymaya.com/sandbox/payments_api/Maya_Bills_Pay_Direct_Biller_postman_collection.zip

Setup

1. Download **Postman** from the link provided above and install it.
2. Download the **Maya Bills Pay Direct Biller Postman Collection** package.
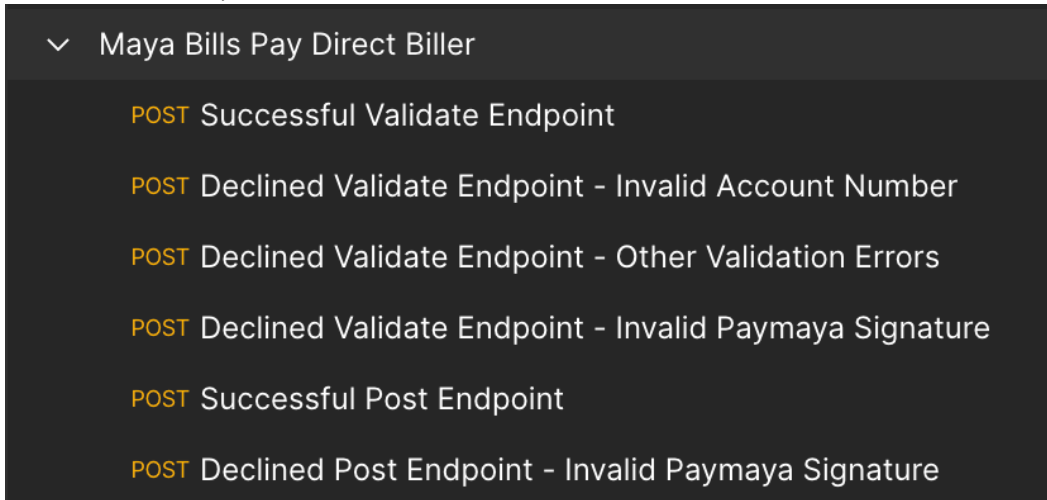3. Create new workspace by clicking **Workspaces** + **New Workspace**
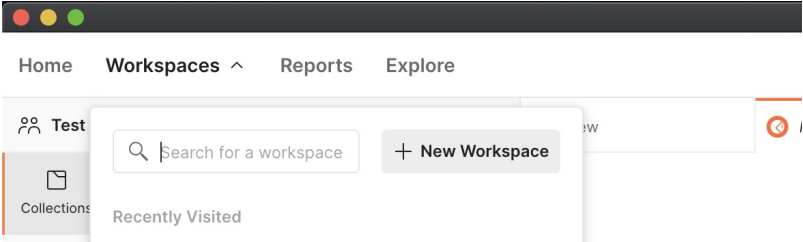
4. In workspace, click *Import* and on the *File* tab, click on *Choose Files*.

5. Select the downloaded *Maya Bills Pay Direct Biller.postman_collection.json* file and click *Import*.
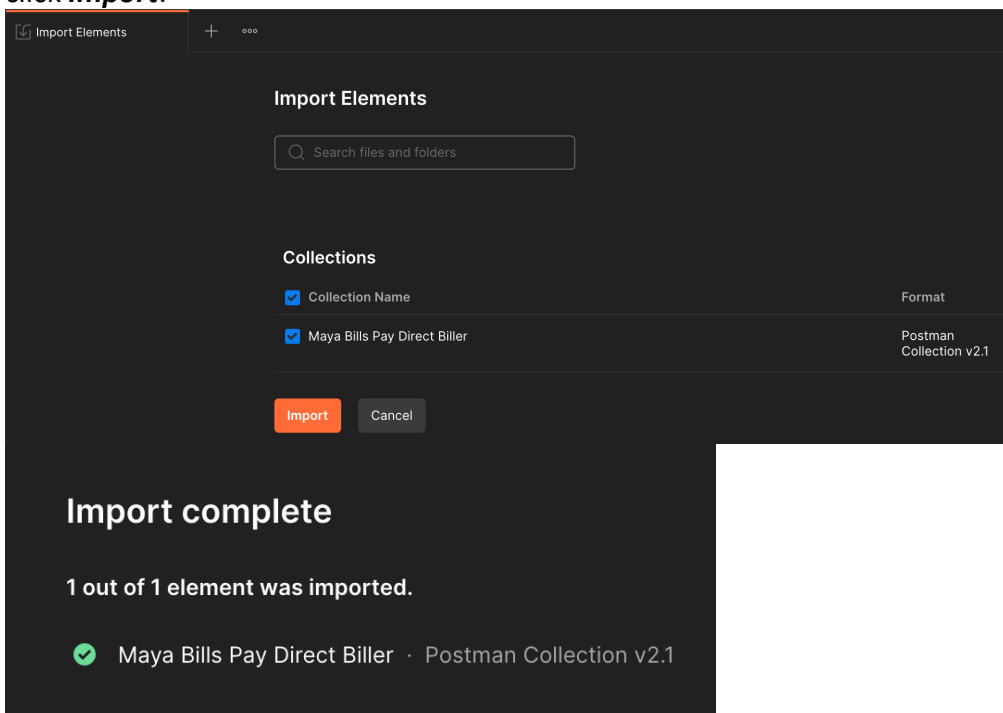
6. When successful, the collection should be added as shown below.

Test Configurations

1. In **Maya Bills Pay Direct Biller** collection, click on Variables tab.
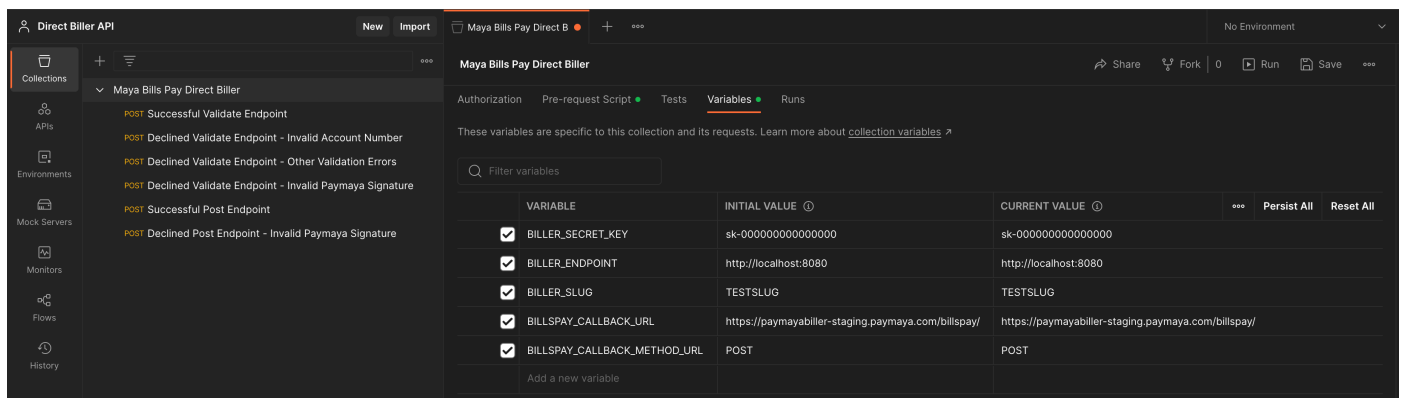
2. Update the following variables in **CURRENT VALUE** column:

Variable	Description	Remarks
BILLER_SECRET_KEY	This is your API secret key	Staging key will be provided during Integration Testing with Maya Bills Pay. You can use a dummy secret key during pre-integration testing of your service. Make sure that BILLER_SECRET_KEY to use in Postman is aligned with the current key configured in your API service.
BILLER_ENDPOINT	This is your API base endpoint.	If your full URL /v1/validate is **https://billspay.company.com/path/v1/validate**, then the value for BILLER_ENDPOINT should be **https://billspay.company.com/path**

		The pre-integration Postman testing tool can be used to test local deployments as well, i.e: **http://localhost:8080**
BILLER_SLUG	This is your biller internal ID.	Biller slug assigned to biller partner would be provided during Integration Testing with Maya Bills Pay. For pre-integration test, a dummy BILLER_SLUG could be used. We would suggest to make your services' BILLER_SLUG be configurable so as to easily switch once the actual BILLER_SLUG has been designated.
BILLER_CALLBACK_URL	This is Maya's callback endpoint.	No need to update for now.
BILLER_CALLBACK_METHOD_URL	This is Maya's callback HTTP method.	No need to update for now.

3. Click on the **SAVE** button once values have been updated to apply the changes made.

No Environment ⌄

    ↗ Share   ⑁ Fork   0   ▶ Run   💾 Save   ⦿⦿⦿

Save collection

ns

ore about collection variables ↗

ⓘ	CURRENT VALUE ⓘ	⦿⦿⦿	**Persist All**	**Reset All**
000000	sk-000000000000000			
:8080	http://localhost:8080			
	TESTSLUG			
abiller-staging.paymaya.com/billspay/	https://paymayabiller-staging.paymaya.com/billspay/			

Testing

Validate Endpoint Testing
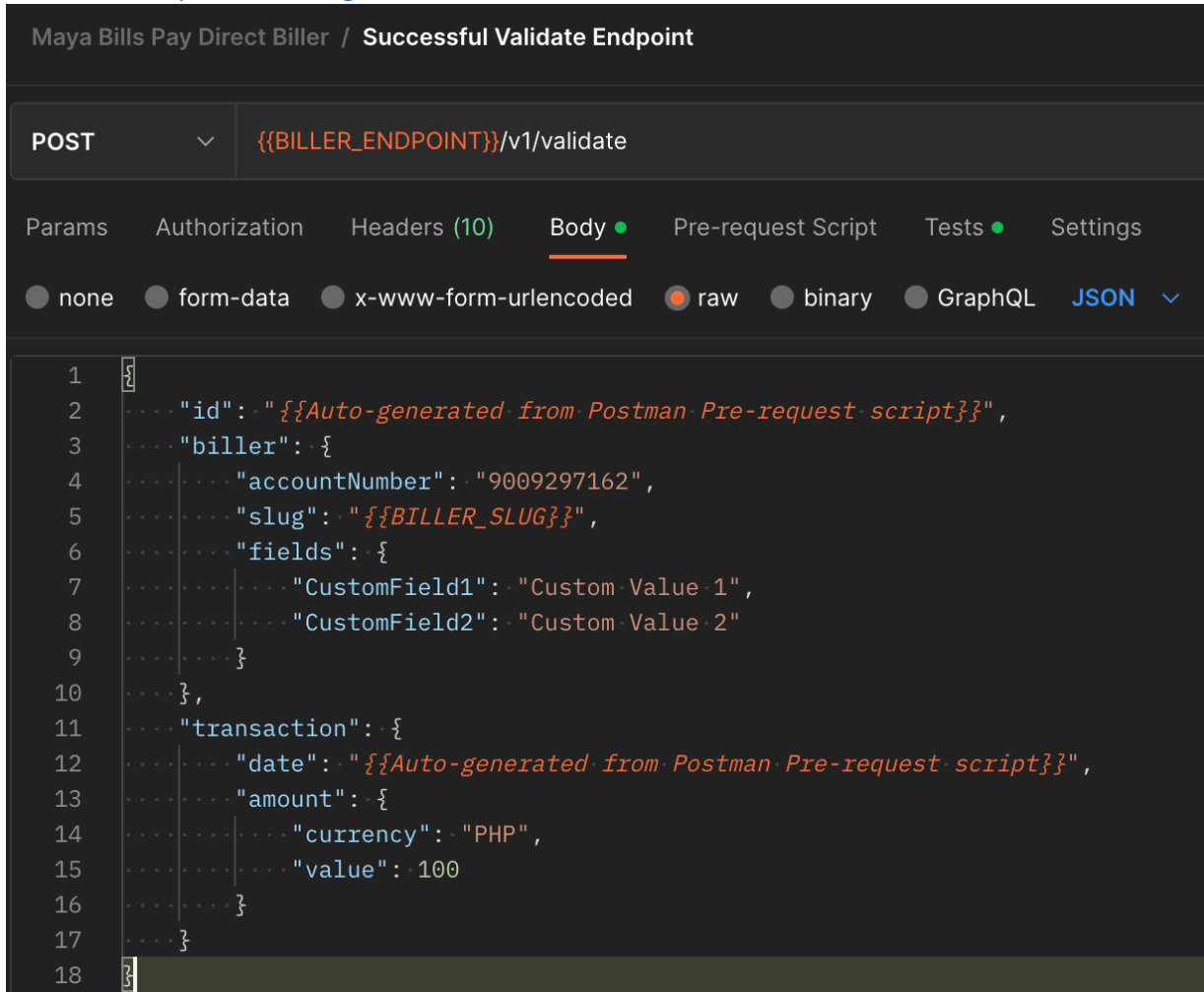
1. For each validate request included in the Postman collection, you need to update the following fields from the JSON body (this should depend on the scenario to test):

Request Field	Required?	Remarks
biller.accountNumber	Yes	Update the value depending on the test scenario to accomplish. Input a valid account number for your successful scenario. Use an invalid account number to test for invalid account number scenario.
biller.fields	No	Update the key value pairs accordingly if you are expecting other fields. If you are not expecting any custom fields, leave the *fields* as a blank node **{}**.
transaction.amount.value	Yes	Update the value to any acceptable value by your service. For Declined scenarios involving amount validation, update as necessary.

2. Click on the **SAVE** button once values have been updated to apply the changes made.
3. Click on the **Send** button to execute the request.
4. Check the bottom part of the Postman window for the response body from your service. Corresponding tests have been provided as well per request to verify if your API's response complies with Maya's Biller API requirements.
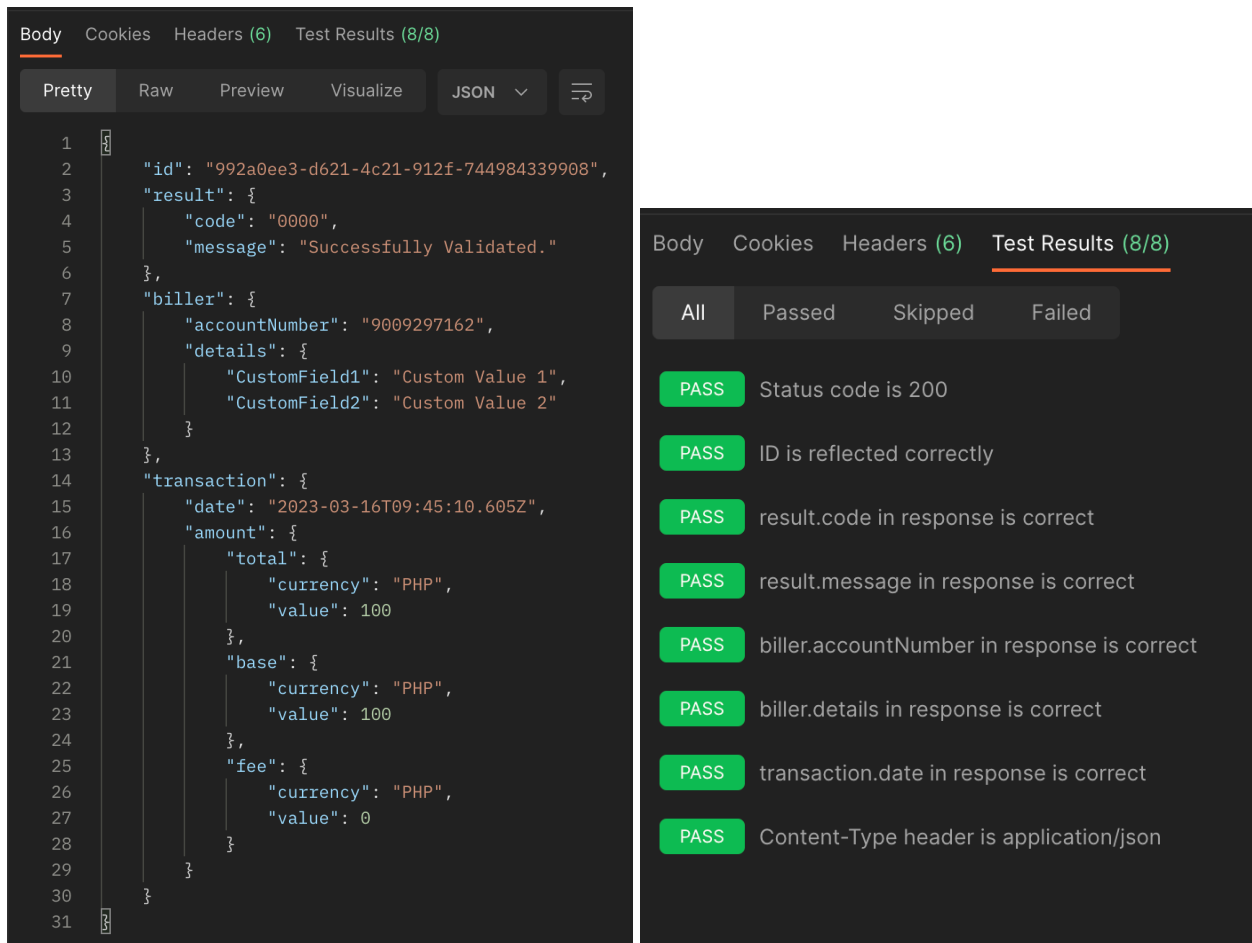
Validate Scenarios

Scenario	Description
Successful Validate Endpoint	Happy path testing for validate endpoint. Input valid account number in **biller.accountNumber** If applicable, input valid/acceptable values under **biller.fields** and **transaction.amount.value** to test all valid scenarios for your API.
Declined Validate Endpoint - Invalid Account Number	Request to test your invalid account number scenario. Input an expected invalid account number in **biller.accountNumber**.
Declined Validate Endpoint - Other Validation Errors	Request to test any other validation error scenarios.

	By default, the Postman collection has a negative value for **transaction.amount.value** as this is an expected validation error for the amount field. If your API has an expected validation logic, update the appropriate fields as necessary. Examples: 1. You have a minimum amount validation for your API of Php 500. Input a value less than 500 under **transaction.amount.value** to test your validation logic. 2. You have to match on your end that an account number inputted belongs to a person with *First Name: Juan* and *Last Name: Dela Cruz*. Update the values under **biller.fields** to test your validation logic for this. For such validation logic that your API might have, we suggest returning a descriptive error message under **result.message** in your validation response. This exact message would be directly displayed in the Maya App and read by the customer trying to pay for the bill. Examples: 1. Amount should be greater than 500. 2. Account Number does not match user on record.
Declined Validate Endpoint - Invalid Paymaya Signature	Request to test if your API has computed the paymaya-signature header correctly.

Post Endpoint Testing


```
Maya Bills Pay Direct Biller  /  Successful Post Endpoint

POST          ∨     {{BILLER_ENDPOINT}}/v1/post

Params   Authorization   Headers (10)   Body •   Pre-request Script   Tests •   Settings

○ none   ○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   ○ GraphQL   JSON ∨

 1  {
 2      "id": "{{Auto-generated from Postman Pre-request script}}",
 3      "biller": {
 4          "accountNumber": "9009297162",
 5          "slug": "{{BILLER_SLUG}}",
 6          "fields": {
 7              "CustomField1": "Custom Value 1",
 8              "CustomField2": "Custom Value 2"
 9          }
10      },
11      "transaction": {
12          "date": "{{Auto-generated from Postman Pre-request script}}",
13          "amount": {
14              "total": {
15                  "currency": "PHP",
16                  "value": 110
17              },
18              "base": {
19                  "currency": "PHP",
20                  "value": 100
21              },
22              "fee": {
23                  "currency": "PHP",
24                  "value": 10
25              }
26          }
27      },
28      "callback": {
29          "url": "{{BILLSPAY_CALLBACK_URL}}",
30          "method": "{{BILLSPAY_CALLBACK_METHOD_URL}}"
31      }
32  }
```

1. For both post requests included in the Postman collection, you need to update the following fields from the JSON body:

Request Field	Required?	Remarks
biller.accountNumber	Yes	Update the value depending on the test scenario to accomplish. Input a valid account number for your successful scenario. Use an invalid account number to test for invalid account number scenario.
biller.fields	No	Update the key value pairs accordingly if you are expecting other fields. If you are not expecting any custom fields, leave the *fields* as a blank node **{}**.

transaction.amount.total.value	Yes	Update the value to any acceptable value by your service. This should be the total amount of the bills payment transaction. i.e: base amount + fee amount
transaction.amount.base.value	Yes	This should be the base amount of the bills payment transaction and should exclude fees.
transaction.amount.fee.value	Yes	This should be the fee amount of the bills payment transaction. The fee amount is the amount the customer needs to pay on top of the bill amount to be paid.

2. Click on the **SAVE** button once values have been updated to apply the changes made.
3. Click on the **Send** button to execute the request.
4. Check the bottom part of the Postman window for the response body from your service. Corresponding tests have been provided as well per request to verify if your API's response complies with Maya's Biller API requirements.

Body   Cookies   Headers (6)   Test Results (9/9)

Pretty   Raw   Preview   Visualize   JSON

```
 1  {
 2      "id": "b78f16a6-c7e4-4a53-8618-ecb80094085d",
 3      "biller": {
 4          "accountNumber": "9009297162",
 5          "details": {
 6              "CustomField1": "Custom Value 1",
 7              "CustomField2": "Custom Value 2"
 8          }
 9      },
10      "transaction": {
11          "date": "2023-03-20T02:03:35.101Z",
12          "receiptNumber": "ecb80094085d",
13          "amount": {
14              "total": {
15                  "currency": "PHP",
16                  "value": 110
17              },
18              "base": {
19                  "currency": "PHP",
20                  "value": 100
21              },
22              "fee": {
23                  "currency": "PHP",
24                  "value": 10
25              }
26          }
27      },
28      "result": {
29          "code": "0000",
30          "message": "Successfully Posted."
31      }
32  }
```

Body   Cookies   Headers (6)   **Test Results (9/9)**

All   Passed   Skipped   Failed

PASS   Status code is 200

PASS   ID is reflected correctly

PASS   result.code in response is correct

PASS   result.message in response is correct

PASS   biller.accountNumber in response is correct

PASS   biller.details in response is correct

PASS   transaction.date in response is correct

PASS   transaction.receiptNumber in response is correct

PASS   Content-Type header is application/json

Post Scenarios

Scenario	Description
Successful Post Endpoint	Happy path testing for post endpoint. Input valid account number in **biller.accountNumber** If applicable, input valid/acceptable values under **biller.fields** and **transaction.amount.total.value/transaction.amount.base.value/ transaction.amount.fee.value** to test all valid scenarios for your API. For this leg, you service should persist the request on your end as these are for posting already. We recommend to persist the callback URL included in the request as well as this is the endpoint to hit to update
Declined Post Endpoint - Invalid Paymaya Signature	Request to test if your API has computed the paymaya-signature header correctly.