

Braintree Connector



Overview	2
Introduction	2
Software Requirements	2
Connector Namespace and Schema	2
Maven Dependency Information	3
Configuration	4
Creating a New Project	4
Configuring the Braintree Global Element	5
Authentication	5
Using Braintree API Credentials	5
Using PayPal Express Checkout Access Token	6
Understanding the Braintree Connector	7
Common Use Cases	8
Create a Customer with Payment Method and View Details	8
Undo a transaction	11
Submit transaction for settlement	13
Recurring Billing Details	19
GraphQL Example for Query and Mutation	27
Resources	32

Overview

Braintree is a full-stack payments platform that makes it easy to accept payments in your app or website. Braintree service replaces the traditional model of sourcing a payment gateway and merchant account from different providers. From single-touch payments to mobile SDKs and foreign currency acceptance, Braintree provides everything you need to start accepting payments today.

The Braintree connector allows you to perform Braintree server operations using Anypoint Platform. The Braintree server operations enables you to act on the collected payment information from customers. It supports various operations of Customer, Payment Method, Transaction and Client Token.

Track features and API version updates using the [Braintree Connector Release Note](#). Review the connector operations and see how they work by reviewing the technical reference alongside the demo applications.

Introduction

This document assumes that user is familiar with [Braintree Payments](#), [Anypoint Connectors](#) and [Anypoint Studio](#). To increase familiarity with Studio, consider completing a [Anypoint Studio Tutorial](#). This page requires basic knowledge of [Mule Key Concepts](#) and [Braintree](#).

Software Requirements

For software requirements, visit the [BrainTree Connector Release Note](#).

Installation

To install the connector in Anypoint Studio use the instructions in [Add Modules to Your Project](#).

Additionally, we recommend to keep Studio up to date with its latest version.

Connector Namespace and Schema

While designing Mule application in Anypoint Studio, when user drag the connector from the palette onto the

Anypoint Studio canvas, studio automatically populates the XML code with the connector namespace and schema location.

Namespace	http://www.mulesoft.org/schema/mule/braintree
Schema Location	http://www.mulesoft.org/schema/mule/braintree/current/mule-braintree.xsd

If you are manually coding the Mule application in Studio's XML editor or another text editor, define the namespace and schema location in the header of your Configuration XML, inside the <mule> tag.

```
<mule
  xmlns:slack="http://www.mulesoft.org/schema/mule/slack"
  xmlns:ee="http://www.mulesoft.org/schema/mule/ee/core"
  xmlns:braintree="http://www.mulesoft.org/schema/mule/braintree"
  xmlns:http="http://www.mulesoft.org/schema/mule/http"
  xmlns="http://www.mulesoft.org/schema/mule/core"
  xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.mulesoft.org/schema/mule/core
  http://www.mulesoft.org/schema/mule/core/current/mule.xsd
  http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-
  http.xsd http://www.mulesoft.org/schema/mule/braintree
  http://www.mulesoft.org/schema/mule/braintree/current/mule-braintree.xsd">
</mule>
```

Note: Use current in the schema path. Studio interprets this to the current Mule version.

Maven Dependency Information

After downloading and installing the connector, following steps make the Braintree connector available inside a Mule application for use and to package the application with connector. If using Anypoint Studio, it will do this automatically. For Maven dependency management, include this XML snippet in the pom.xml file in the Mule project.

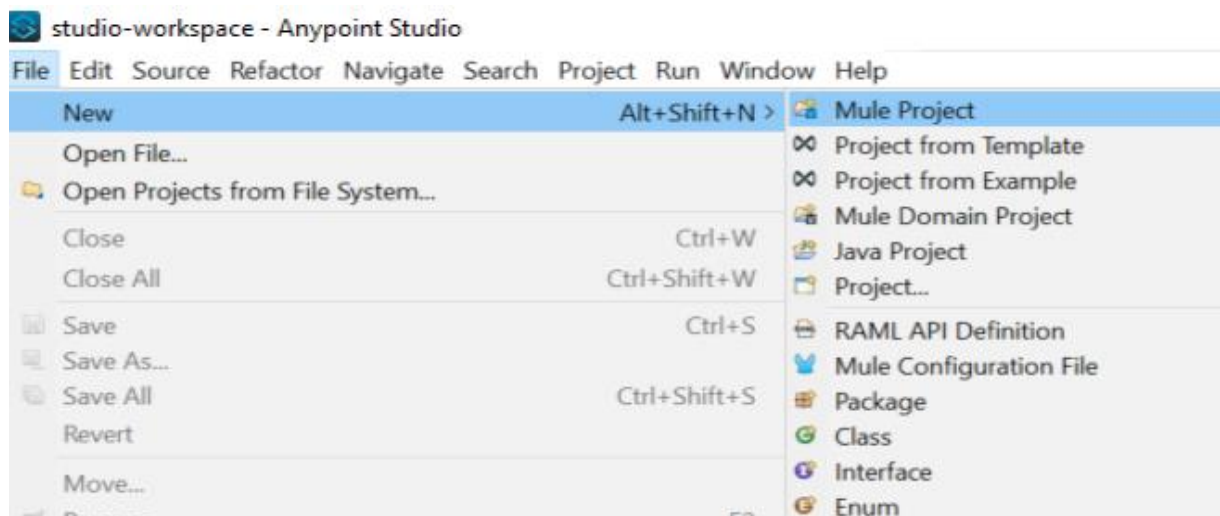
```
<dependency>
  <groupId>com.mulesoft.connectors</groupId>
  <artifactId>mule-braintree-connector</artifactId>
  <version>2.0.0</version>
  <classifier>mule-plugin</classifier>
</dependency>
```

Configuration

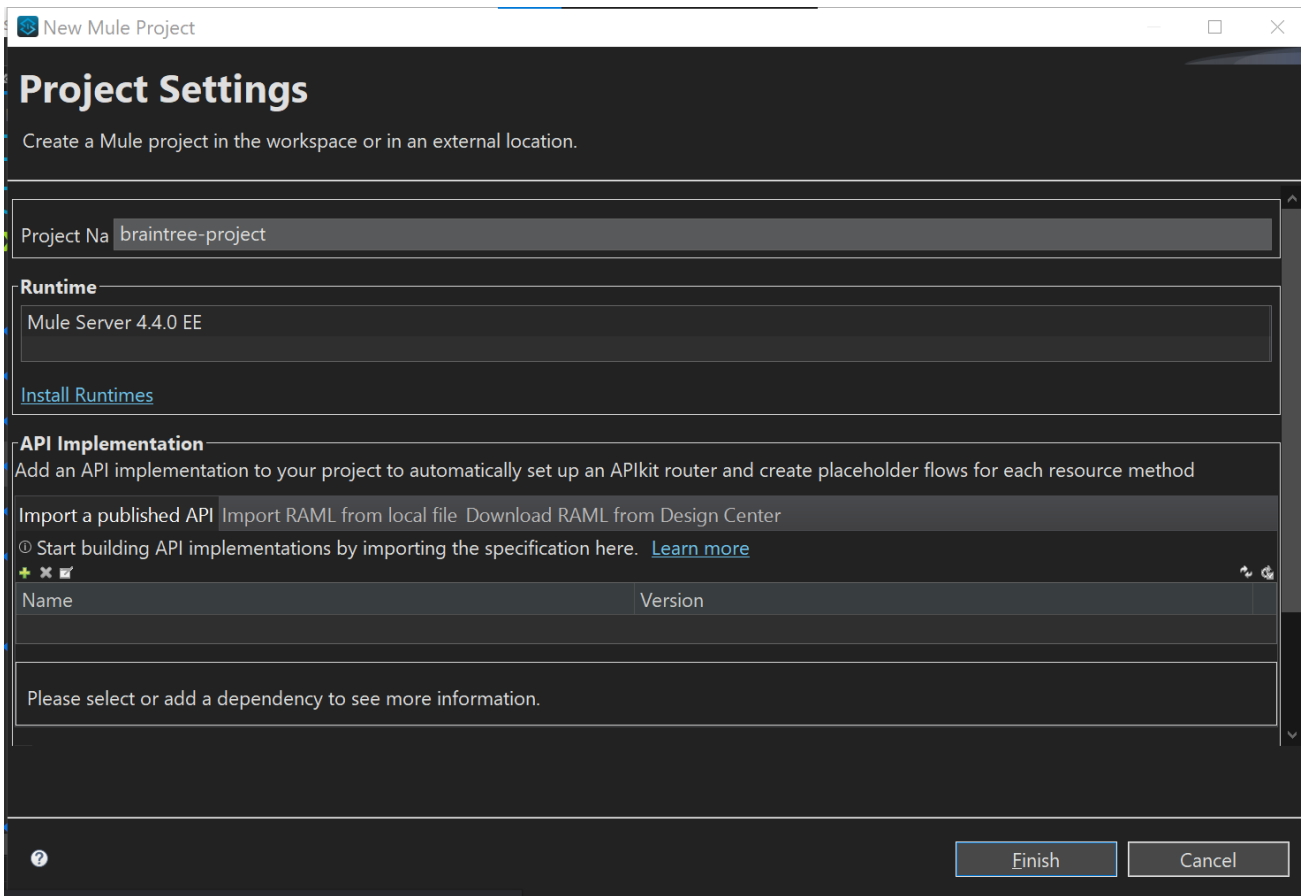
Creating a New Project

To use the Braintree connector in a Mule application project:

1. In Anypoint Studio, click File > New > Mule Project. Select Mule project from the dialog box



2. Enter project name and specify Runtime, API implementation and Project Location if needed.



3. Click **Finish** to create the project

Configuring the Braintree Global Element

To use the Braintree connector in the Mule application, users must configure a global Braintree element that is used by the Braintree connector to authenticate. The Braintree connector supports API credentials authentication as well as PayPal Express Checkout Access Token configuration option.

Authentication

Using Braintree API Credentials

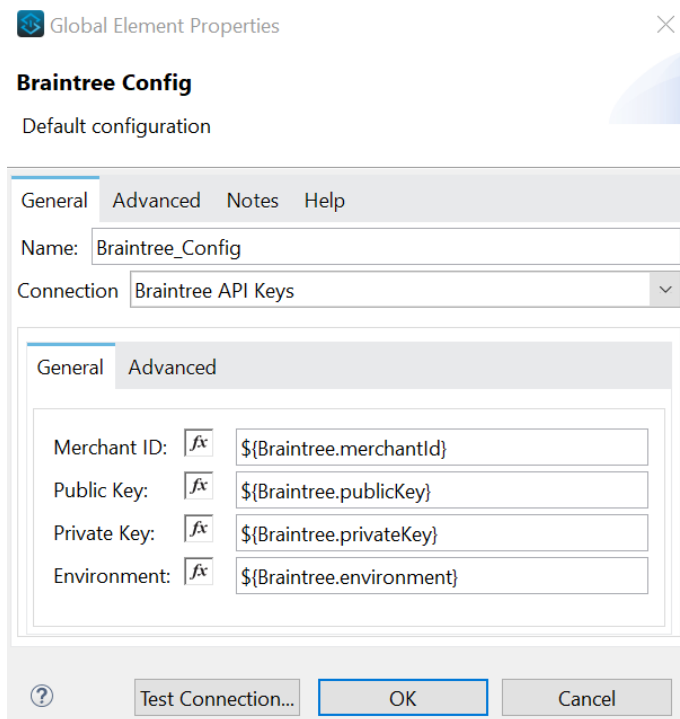
To access the Braintree server, the user needs to configure API credentials as mentioned below.

API credentials are unique account identifiers that must be used to configure the Braintree connector. For creating and managing API Credentials, refer [Braintree Gateway Credentials](#).

Following parameters are required for **API CREDENTIALS** configuration:

Field	Description
Merchant ID	Merchant ID is the unique identifier for your entire gateway account, including the multiple merchant accounts that may be in your gateway.
Public key	This is a user-specific public identifier. Each user associated with your Braintree gateway will have their own public key.
Private key	This is a user-specific private identifier. Each user associated with your Braintree gateway will have their own private key.
Environment	The environment specifies where requests via the API should be directed. e.g sandbox, production.

Please select the “Braintree API Keys” option from “Connection” parameter dropdown.



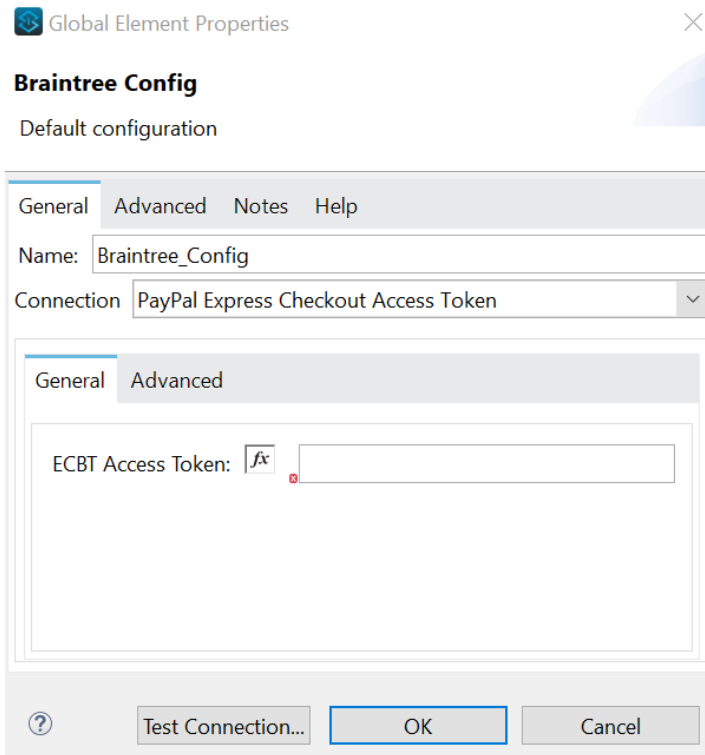
Using PayPal Express Checkout Access Token

PayPal Express Checkout via the Braintree SDK integration allows the merchant to create (credentials in the form of) an access token via their PayPal account. This access token can be used instead of Braintree API credentials for Braintree API calls.

To get an express checkout access token, log into sandbox business dashboard (<https://developer.paypal.com/developer/applications>), under My Apps & Credentials, see the section Express Checkout via Braintree SDK - Sandbox Accounts, and you can create a sandbox access token.

For this please select the “PayPal Express Checkout Access Token” option from the “Connection” parameter dropdown.

Field	Description
ECBT Access Token	PayPal Express Checkout Access Token generated through PayPal account.



Understanding the Braintree Connector

The Braintree connector functions within a Mule application. Braintree connector allows you to perform Braintree server operations through mule applications. Building an application that connects with Braintree user doesn't have to go through the effort of custom-coding (and securing!) a connection. Rather, just drop a connector into the Mule flow, configure a few connection details and begin application running on Braintree.

Braintree Connector is used to connect to the Braintree server. Braintree Connector is built using braintree-java Java API client provided by Braintree. It executes operations exposed by Braintree using configured parameters.

The Braintree connector is configured to authorize access to Braintree server resources. It requires either Public key, Private key, and Merchant ID or PayPal Express Checkout Access for the authentication.

The real value of the Braintree connector is in the way you use it at design-time in conjunction with other

functional features available in Mule.

- **DataSense:** DataSense extracts metadata for Braintree standard response to automatically determine the data type and format that your application must deliver to, or can expect from, Braintree server. Mule does the heavy lifting of discovering the type of data you must send to, or be prepared to receive from Braintree server.
- **Transform Message Component:** This component's integrated scripting language called DataWeave can automatically extract response metadata that you can use to visually map and/or transform to a different data format or structure. Essentially, DataWeave lets you control the mapping between data types. For example, if you configure a Braintree connector in your application, then drop a Transform Message component after the connector, the component uses DataWeave to gather information that DataSense extracted to pre-populate the input values for mapping. In other words, DataSense makes sure that DataWeave knows the data format and structure it must work with so you don't have to figure it out manually. Mule developers can refer to [Braintree Connector API docs](#) and [Braintree Documentation](#) for more information about complex input or return types.

Common Use Cases

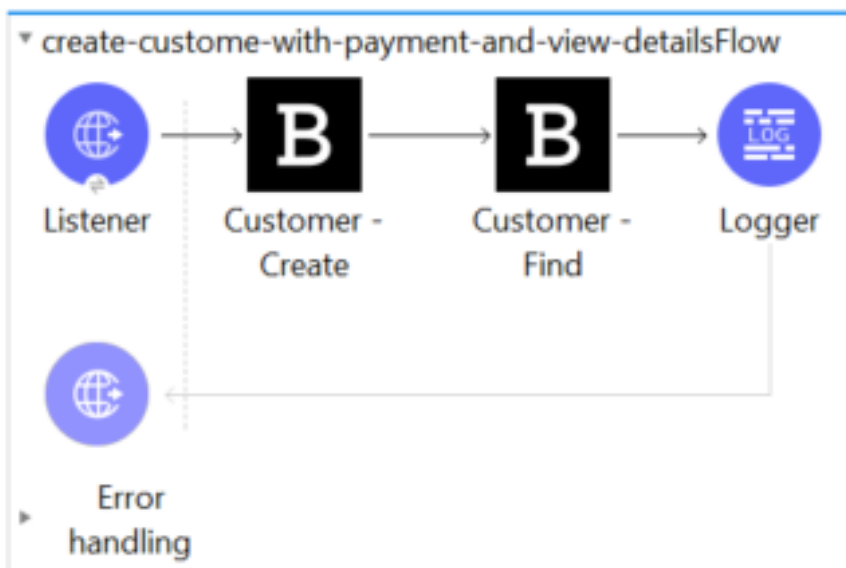
- Create a Customer with Payment Method and View Details
- Undo a transaction
- Submit transaction for settlement
- Recurring Billing Details
- GraphQL Example for Query and Mutation

Create a Customer with Payment Method and View Details

This use case demonstrates the creation of a customer. It creates a customer with customer details, payment method and payment billing address. Then, it gets the customer details by using Customer - Find operation.

- In Anypoint Studio, click **File > New > Mule Project**, name the project and click Finish.
- In the search field, type **Http** and drag the **HTTP Listener** to the canvas.
- Click the **HTTP Listener**, click the green plus sign to the right of Connector Configuration. On the next screen, add the host and port, click OK
- Provide **Path** to HTTP Listener (Example: /customer)

- On the Mule Palette, search for Braintree **Customer - Create** operation and drag it onto the canvas. Provide Customer details.
- Click on the Payment **Method** tab in Customer - Create configuration. Provide payment method details.
- Click on **Payment Method Billing Address** tab in Customer - Create configuration. Provide payment method billing address details.
- On the Mule Palette, search for **Customer - Find** operation and drag it onto canvas.
- Set the 'Customer Id' to **payload.target.id** as expression
- Drag logger onto canvas and log payload as expression.
- After creating the flow, right-click the project and click **Run As > Mule Application**.
- Once the application is deployed, use **HTTP Listener's Path** to execute the Mule Flow.



Use Case Configuration XML : Paste this XML code into Anypoint Studio to experiment with the flow described above.

```

<?xml version="1.0" encoding="UTF-8"?>
<mule xmlns:braintree="http://www.mulesoft.org/schema/mule/braintree"
  xmlns:ee="http://www.mulesoft.org/schema/mule/ee/core"
  xmlns:http="http://www.mulesoft.org/schema/mule/http"
  xmlns="http://www.mulesoft.org/schema/mule/core"
  xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
  
```

http://www.mulesoft.org/schema/mule/ee/core http://www.mulesoft.org/schema/mule/ee/core/current/mule-ee.xsd
http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd

```
http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd
http://www.mulesoft.org/schema/mule/braintree
http://www.mulesoft.org/schema/mule/braintree/current/mule-braintree.xsd">
  <http:listener-config name="HTTP_Listener_config"
    doc:name="HTTP Listener config" doc:id="a3cfe521-b7b9-4a38-8048-ff1f02bf655d">
    <http:listener-connection host="0.0.0.0"
      port="8081" />
  </http:listener-config>
  <configuration-properties doc:name="Configuration properties" doc:id="5380985a-03a1-4a72-94dc-
d32c69578389" file="mule-application.properties" />
  <braintree:config name="Braintree_Config" doc:name="Braintree Config"
    doc:id="358f0686-82fd-438d-93a6-e58a719c361e">
    <braintree:connection merchantId="{Braintree.merchantId}"
      publicKey="{Braintree.publicKey}" privateKey="{Braintree.privateKey}"
      environment="{Braintree.environment}" />
  </braintree:config>
  <flow name="create-custome-with-payment-and-view-detailsFlow"
    doc:id="c97819d2-9e8c-4037-a385-3d40898df5ea">
    <http:listener doc:name="Listener"
      doc:id="5bf1b916-c6c5-43d3-870f-d09e76ec537f" config-ref="HTTP_Listener_config"
      path="/customer" />
    <braintree:create-customer doc:name="Customer - Create"
      doc:id="62e6681a-935c-4e03-b748-623ea0846142" customerFirstName="{customer.name.first}"
      customerLastName="{customer.name.last}" company="{customer.company}"
      fax="{customer.fax}"
      phone="{customer.phone}" website="{customer.website}" email="{customer.email}"
      nonce="fake-valid-nonce" config-ref="Braintree_Config">
    <braintree:payment-method cardholderName="{payment.card.name}"
      number="{payment.card.number}" cvv="{payment.card.cvv}"
      expirationMonth="05" expirationYear="12" deviceData="{payment.device.data}" />
    <braintree:payment-method-options />
    <braintree:address company="{payment.address.company}"
      countryCodeAlpha2="{payment.address.country.alpha2}"
      countryCodeAlpha3="{payment.address.country.alpha3}"
      countryCodeNumeric="{payment.address.country.numeric}"
      countryName="{payment.address.country.name}"
      streetAddress="{payment.address.streetAddress}" region="{payment.address.region}"
      extendedAddress="{payment.address.extendedAddress}" locality="{payment.address.locality}"
      postalCode="{payment.address.postalCode}" />
    <braintree:custom-fields />
  </braintree:create-customer>
  <braintree:find-customer doc:name="Customer - Find" doc:id="f78fa1d2-b788-47d3-8635-
ee5ff15c6d63" config-ref="Braintree_Config" customerId="#[payload.target.id]"/>
  <logger level="INFO" doc:name="Logger" doc:id="06f49150-631d-4dc2-b36c-2d7c16554e08"
    message="#[payload]" />
  </flow>
</mule>
```

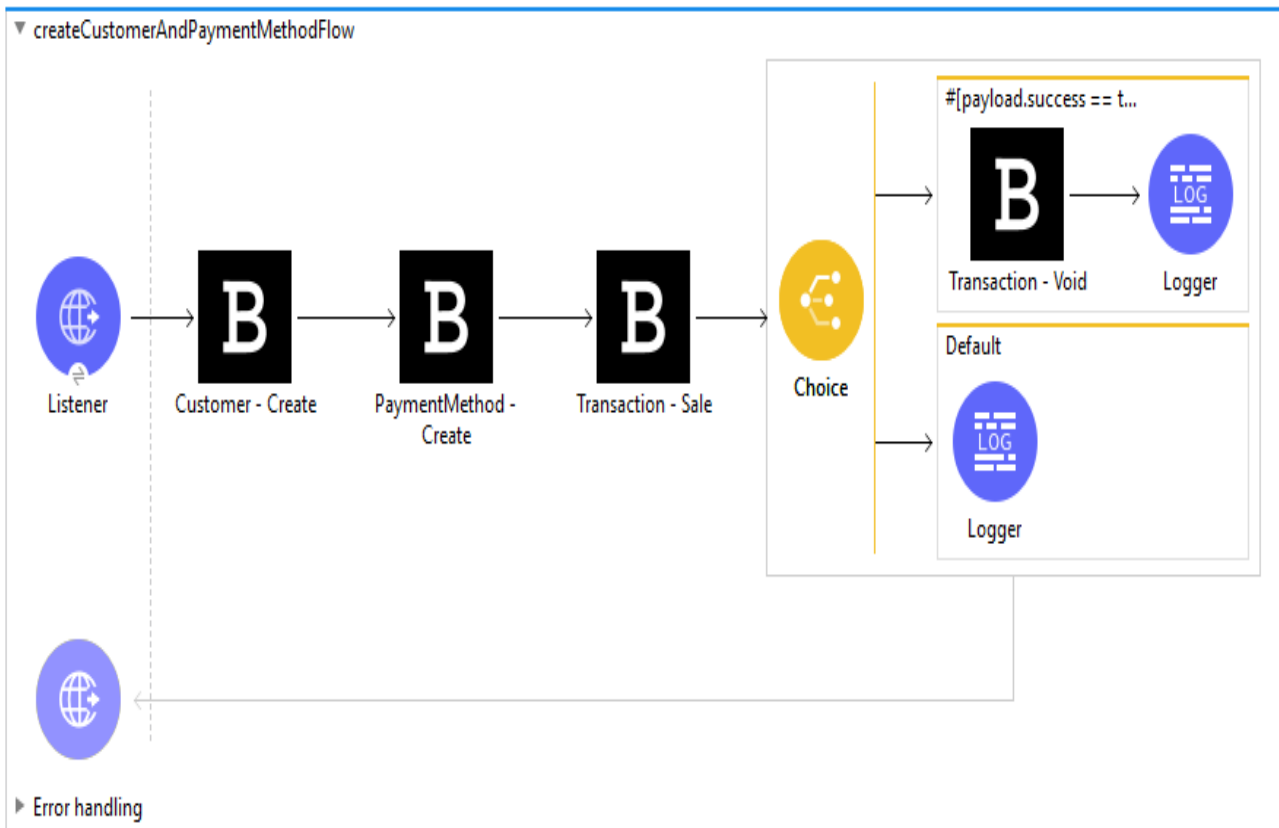
Undo a transaction

This use case demonstrates creating a Customer and Payment Method, perform Transaction then void the transaction before it settles.

- In Anypoint Studio, click **File > New > Mule Project**, name the project, and click **Finish**.
- In the search field, type **Http** and drag the **HTTP Listener** to the canvas.
- Click the **HTTP Listener**, click the **green plus** sign to the right of Connector Configuration. On the next screen, add the host and port, click OK
- Provide **Path** to HTTP Listener (Example: /void)
- On the Mule Palette, search for Braintree **Customer - Create** operation and drag it onto the canvas. Provide Customer details.
- On the Mule Palette, search for Braintree **PaymentMethod – Create** operation and drag it onto the canvas.
- Provide customer ID **payload.target.id** as expression to link this payment method with above customer.
- Provide 'Payment Method Nonce' as **fake-valid-nonce**.
- On the Mule Palette, search for Braintree **Transaction - Sale** operation and drag it onto the canvas.
- In Sale Transaction configuration, provide **Sale Amount** (Example: 100) and Customer ID as expression **payload.target.customerId**.
- Add **Choice** from Mule Palette to the canvas. It is used to switch between the different choices depending on provided expression.
- Add below condition as expression in **Choice** to check if sale transaction operation is a success.

```
payload.success == true
```

- To the expression side, add **Transaction - Void** to undo a transaction with transaction ID as expression **payload.target.id**.
- Add Logger to log the status of **Transaction - Void** operation.
- To the default side of the **Choice**, add the **Logger** to log **"Transaction was unsuccessful"**.
- After creating the flow, right-click the project and click **Run As > Mule Application**.
- Once the application is deployed, use **HTTP Listener Path** to execute the Mule Flow.



Use Case Configuration XML : Paste this XML code into Anypoint Studio to experiment with the flow described above.

```
<?xml version="1.0" encoding="UTF-8"?>
<mule xmlns:braintree="http://www.mulesoft.org/schema/mule/braintree"
  xmlns:http="http://www.mulesoft.org/schema/mule/http"
  xmlns="http://www.mulesoft.org/schema/mule/core"
  xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.mulesoft.org/schema/mule/core
http://www.mulesoft.org/schema/mule/core/current/mule.xsd
http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd
http://www.mulesoft.org/schema/mule/braintree
http://www.mulesoft.org/schema/mule/braintree/current/mule-braintree.xsd">

  <configuration-properties file="mule-application.properties" />
  <http:listener-config name="HTTP_Listener_config"
    doc:name="HTTP Listener config" doc:id="6f6eaa22-7518-4411-9c84-a17a40e91b1d">
    <http:listener-connection host="0.0.0.0"
      port="8081" />
  </http:listener-config>
  <braintree:config name="Braintree_Config" doc:name="Braintree Config"
    doc:id="d3788405-ab3f-4c9c-9de0-3fdb64f261ec">
    <braintree:connection merchantId="{braintree.merchantId}"
```

```

    publicKey="{braintree.publicKey}" privateKey="{braintree.privateKey}"
    environment="{braintree.environment}" />
</braintree:config>
<flow name="createCustomerAndPaymentMethodFlow" doc:id="d968819e-b2f9-443f-b9e5-
3430439607e3">
  <http:listener doc:name="Listener"
    doc:id="84d31ca0-e735-47cd-b4fd-48bf4fc60495" path="{braintree.httpListener.path}"
    config-ref="HTTP_Listener_config" />
  <braintree:create-customer doc:name="Customer - Create"
    doc:id="bbb88ddb-1127-4a88-8211-b83a4f780a76" customerFirstName="{customer.firstName}"
    customerLastName="{customer.lastName}" company="{customer.company}"
    fax="{customer.fax}" phone="{customer.phone}" website="{customer.website}"
    email="{customer.email}" config-ref="Braintree_Config" />
  <braintree:create-payment-method
    doc:name="PaymentMethod - Create" doc:id="697f126b-7bff-4e39-a601-85bc0bb42b7e"
    config-ref="Braintree_Config" customerId="#[payload.target.id]"
    paymentMethodNonce="{paymentMethod.nonce}" />

  <braintree:sale doc:name="Transaction - Sale"
    doc:id="fc4b68e7-5b79-46a0-ba5e-25396c09f869" config-ref="Braintree_Config"
    saleAmount="{transaction.amount}">
    <braintree:sale-customer>
      <braintree:existing-customer customerId="#[payload.target.customerId]" />
    </braintree:sale-customer>
  </braintree:sale>

  <choice doc:name="Choice" doc:id="9522224f-3cb0-4d72-aedd-26976684a68b">
    <when expression="#[payload.success == true]">
      <braintree:void-transaction doc:name="Transaction - Void"
        doc:id="ee7607b4-267b-4d42-9354-89e06cdb177f" config-ref="Braintree_Config"
        transactionId="#[payload.target.id]" />
      <logger level="INFO" doc:name="Logger"
        doc:id="2b3d98ea-9aa5-44ff-ab92-d84486ec0d6c" message="#[payload]" />
    </when>
    <otherwise>
      <logger level="INFO" doc:name="Logger"
        doc:id="3a90544a-2a61-4dca-bed7-f523627800d6" message="#["Transaction
unsuccessful."]' />
    </otherwise>
  </choice>
</flow>
</mule>

```

Submit transaction for settlement

This use case demonstrates creating a sale transaction and submit it for settlement/partial settlement.

- In Anypoint Studio, click File > **New** > **Mule Project**, name the project, and click **Finish**.

- In the search field, type **Http** and drag the **HTTP Listener** to the canvas.
- Click the **HTTP Listener**, click the **green plus** sign to the right of Connector Configuration. On the next screen, add the host and port, click **OK**
- Provide **Path** to HTTP Listener (Example: /settleTransaction)
- This mule application allows users to do **Submit for Partial Settlement** or **Submit for Settlement** based on a flag **isPartialSettlement**.
- Add **Set Variable** and set variable name as **isPartialSettlement** and value as **true** if you want to do Partial Settlement, **false** otherwise.
- Add **Flow Reference** to call another sub flow '**createSaleTransactionFlow**' to create a sale transaction.
- Add **Logger** to log the response of the flow.
- Create a new sub flow **createSaleTransactionFlow** to create a sale transaction.
- On the Mule Palette, search for Braintree **Transaction - Sale** operation and drag it onto the canvas.
- Provide **Sale Amount** (Example- 500).
- Provide Customer details like first name, last name, email, phone etc.
- Provide Payment Method details like card holder name, number, CVV, expiration month, expiration year, threeDSecureRequired etc.
- Add **Choice** from Mule Palette to the canvas. It is used to switch between the different choices depending on the expression provided.
- Add below condition as expression in **Choice** to check if sale transaction operation is a success.

```
payload.success == true
```

- To the expression side, add **Set Variable** with variable name as **transactionID** and value as expression **payload.target.id**. This transaction ID is later used by **Transaction - Find** to get the status of the transaction.
- Add **Flow Reference** to call another sub flow '**submitTransactionForSettlementFlow**' to submit transactions for settlement.
- To the default side of the **Choice**, add the **Logger** to log "**Transaction was unsuccessful. Retrying creating a sale transaction..**".
- Add Flow Reference to call the same sub flow '**createSaleTransactionFlow**' again.
- Create a new sub flow **submitTransactionForSettlementFlow**.
- It submits transactions for settlement/partial settlement based on the value of **isPartialSettlement**

variable.

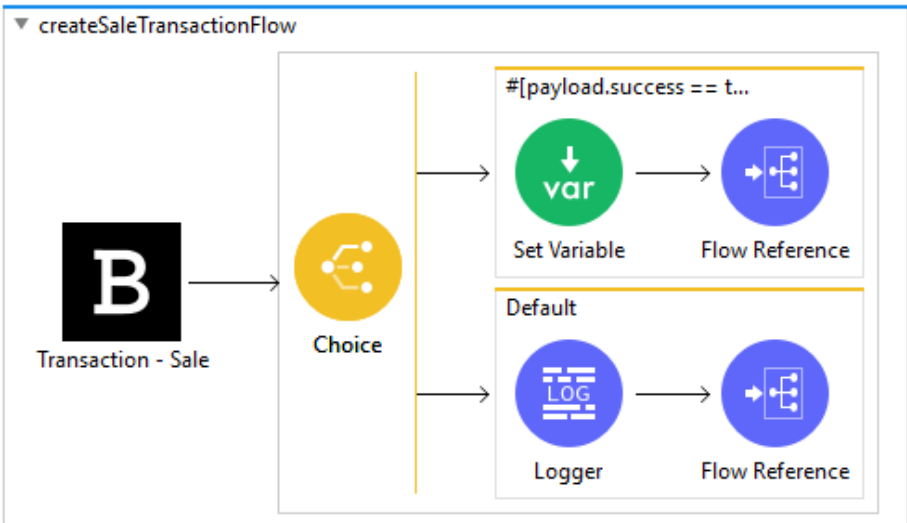
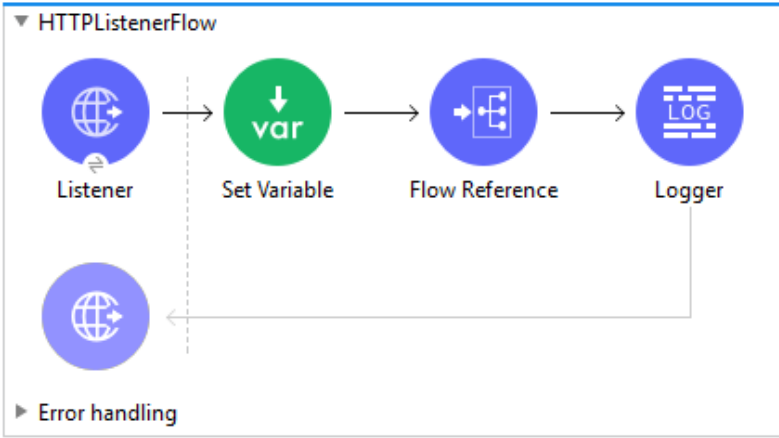
- Add **Transaction - Find** to the canvas and provide transaction ID as expression **vars.transactionID**. This operation is used to get the status of the transaction.
- Add **Set Variable** and set variable name as **transactionStatus** and value as expression **payload.status**.
- Add **Choice** from Mule Palette to the canvas. It is used to switch between the different choices depending on the provided expression.
- Add below condition as expression in **Choice** to check if transaction is AUTHORIZED and you want to do full settlement.

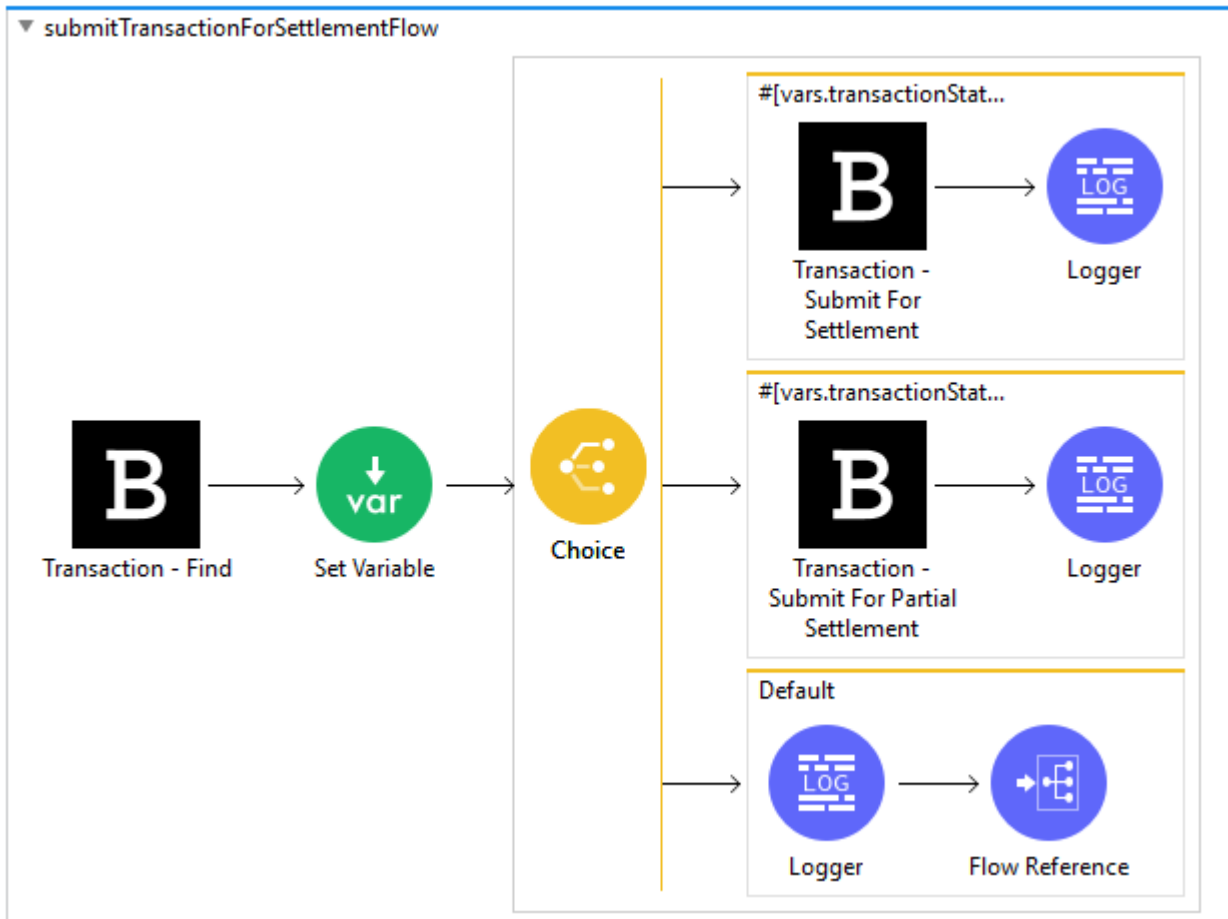
```
vars.transactionStatus == 'AUTHORIZED' and vars.isPartialSettlement == 'false'
```

- Add **Submit for Settlement** operation to the canvas. Provide **Transaction ID** as expression **vars.transactionID**.
- Add **Logger** with appropriate message.
- Add below condition as expression in **Choice** to check if transaction is AUTHORIZED and you want to do partial settlement.

```
vars.transactionStatus == 'AUTHORIZED' and vars.isPartialSettlement == 'true'
```

- Add **Submit for Partial Settlement** operation to the canvas. Provide **Transaction ID** as expression **vars.transactionID**.
- Add **Logger** with appropriate message.
- To the default side of the **Choice**, add the Logger to log "**Waiting for transaction status to be AUTHORIZED**".
- Add Flow Reference to call the same sub flow '**submitTransactionForSettlementFlow**' again.
- After creating the flow, right-click the project and click **Run As > Mule Application**.
- Once the application is deployed, use **HTTP Listener Path** to execute the Mule Flow.





Use Case Configuration XML : Paste this XML code into Anypoint Studio to experiment with the flow described above.

```
<?xml version="1.0" encoding="UTF-8"?>
<mule xmlns:braintree="http://www.mulesoft.org/schema/mule/braintree"
  xmlns:http="http://www.mulesoft.org/schema/mule/http"
  xmlns="http://www.mulesoft.org/schema/mule/core"
  xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.mulesoft.org/schema/mule/core
    http://www.mulesoft.org/schema/mule/core/current/mule.xsd
    http://www.mulesoft.org/schema/mule/http
      http://www.mulesoft.org/schema/mule/http/current/mule-
      http.xsd
    http://www.mulesoft.org/schema/mule/braintree
    http://www.mulesoft.org/schema/mule/braintree/current/mule-braintree.xsd">
  <configuration-properties file="mule-application.properties" />

  <http:listener-config name="HTTP_Listener_config"
    doc:name="HTTP Listener config" doc:id="2d1452bf-3d8f-423a-9bfa-296a215f7f11">
    <http:listener-connection host="0.0.0.0"
      port="8081" />
  </http:listener-config>
</mule>
```

```

</http:listener-config>

<brantree:config name="Braintree_Config" doc:name="Braintree Config"
  doc:id="51a76243-e2a9-4b30-ac0e-1b0fe91720cd">
  <brantree:connection merchantId="{braintree.merchantId}"
    publicKey="{braintree.publicKey}" privateKey="{braintree.privateKey}"
    environment="{braintree.environment}" />
</braintree:config>

<flow name="HTTPListenerFlow" doc:id="cb25ae51-72da-4daf-ba73-7431ad88cd8a">
  <http:listener doc:name="Listener"
    doc:id="9d40d887-67e0-4578-842b-2cab68eb7b7a" path="/settleTransaction"
    config-ref="HTTP_Listener_config" />
  <set-variable value="{braintree.isPartialSettlement}"
    doc:name="Set Variable" doc:id="18388922-266f-4360-bec2-8097543caf95"
    variableName="isPartialSettlement" />
  <flow-ref doc:name="Flow Reference" doc:id="9c7c686b-1e41-4799-96d5-8be84a24791b"
    name="createSaleTransactionFlow" />
  <logger level="INFO" doc:name="Logger"
    doc:id="1b91f524-6993-420e-b794-15c9dfc226e2" message="#[payload]" />
</flow>

<sub-flow name="createSaleTransactionFlow" doc:id="df75d219-101d-46a7-8787-44ebbea4a619">
  <brantree:sale doc:name="Transaction - Sale"
    doc:id="108a09d1-ec8f-4195-a090-b1497bd3a692" saleAmount="{transaction.sale.amount}"
    config-ref="Braintree_Config">
  <brantree:sale-customer>
    <brantree:new-customer customerFirstName="{customer.firstName}"
      customerLastName="{customer.lastName}" email="{customer.email}"
      phone="{customer.phone}" />
  </braintree:sale-customer>
  <brantree:sale-payment-options>
    <brantree:card-details cardholderName="{paymentMethod.cardholderName}"
      number="{paymentMethod.number}" cvv="{paymentMethod.cvv}"
      expirationMonth="{paymentMethod.expirationMonth}"
      expirationYear="{paymentMethod.expirationYear}" />
  </braintree:sale-payment-options>
  <brantree:sale-options />
  <brantree:custom-fields />
</braintree:sale>
  <choice doc:name="Choice" doc:id="4b79dd5d-bd83-41b9-aeb8-309c6939bc80">
    <when expression="#[payload.success == true]">
      <set-variable value="#[payload.target.id]" doc:name="Set Variable"
        doc:id="dde1650b-b8c8-4a79-b26b-4488dbd46ed2" variableName="transactionID" />
      <flow-ref doc:name="Flow Reference" doc:id="6d43fbba-543f-4acb-9c6e-1f23ef424a9f"
        name="submitTransactionForSettlementFlow" />
    </when>
    <otherwise>
      <logger level="INFO" doc:name="Logger"
        doc:id="8ddcd4b1-d005-48fd-9a38-27087416a568" message="#["Transaction
        unsuccessful. Retrying.."]" />
      <flow-ref doc:name="Flow Reference" doc:id="35eec548-21ec-458b-bf74-9767a10b429c"

```

```

        name="createSaleTransactionFlow" />
    </otherwise>
</choice>
</sub-flow>

<sub-flow name="submitTransactionForSettlementFlow" doc:id="1b3df2be-f583-4ecd-8733-
b0490873ab04">
    <brantree:find-transaction doc:name="Transaction - Find"
        doc:id="d52f47d7-69a2-483b-9f07-65c5ab76106b" config-ref="Braintree_Config"
        transactionId="#[vars.transactionID]" />
    <set-variable value="#[payload.status]" doc:name="Set Variable"
        doc:id="5db644e3-a72e-440a-b48c-5031648b0934" variableName="transactionStatus" />

    <choice doc:name="Choice" doc:id="007c9565-e2f6-4a4a-8e90-12881402f51f">
        <when
            expression="#[vars.transactionStatus == 'AUTHORIZED' and vars.isPartialSettlement ==
'false']">
            <brantree:submit-for-settlement
                doc:name="Transaction - Submit For Settlement" doc:id="d89abc23-d78f-4421-a498-
86d216b9f29e"
                config-ref="Braintree_Config" transactionId="#[vars.transactionID]" />
            <logger level="INFO" doc:name="Logger"
                doc:id="64728baa-b558-4c7d-8719-1e18f24a8485" message="#['Submitted for Settlement.
Amount - 500 USD']" />
            </when>
            <when
                expression="#[vars.transactionStatus == 'AUTHORIZED' and vars.isPartialSettlement == 'true']">
                <brantree:submit-for-partial-settlement
                    doc:name="Transaction - Submit For Partial Settlement" doc:id="28e9cbc1-2ddf-4d9a-abcf-
79ece0a9f5d1"
                    config-ref="Braintree_Config" authorizedTransaction="#[vars.transactionID]"
                    amount="{transaction.partialSettlement.amount}" />
                <logger level="INFO" doc:name="Logger"
                    doc:id="3aab3d76-7f97-41bd-a38d-a7ebd25f7cae" message="#['Submitted for Partial
Settlement. Amount - 200 USD']" />
                </when>
            <otherwise>
                <logger level="INFO" doc:name="Logger"
                    doc:id="2f212901-66d0-44ff-b7f1-db166719a96e" message="#['Waiting for transaction status
to be AUTHORIZED']" />
                <flow-ref doc:name="Flow Reference" doc:id="fbfaadf5-ec9d-4b15-9242-f07461b139c2"
                    name="submitTransactionForSettlementFlow" />
            </otherwise>
        </choice>
    </sub-flow>
</mule>

```

Recurring Billing Details

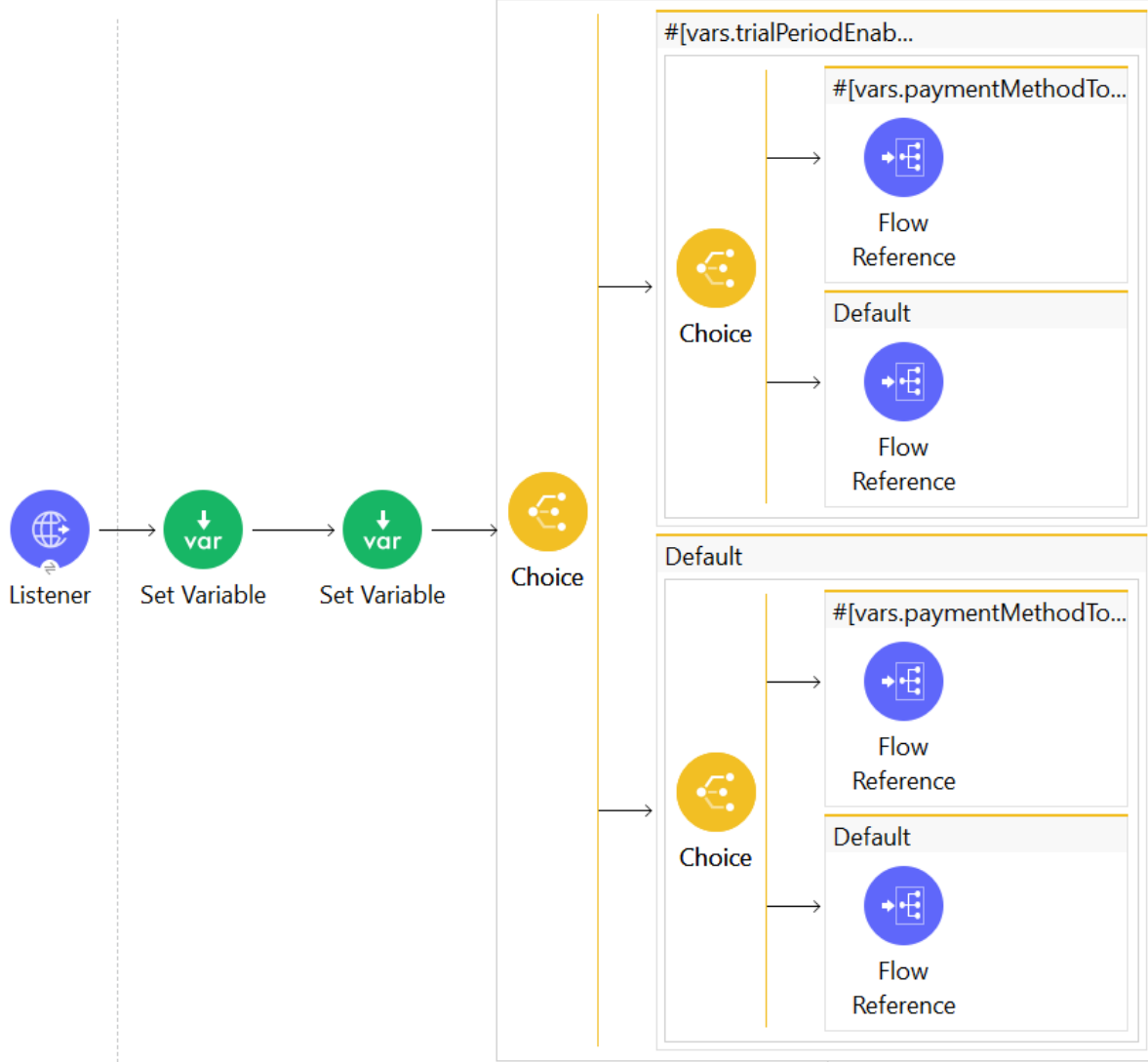
This use case demonstrates recurring billing functionality by showing the creation of plan as well as

subscription with and without trial period. In Recurring Billing, Plan is created first if it is required otherwise existing plan can be used for Subscription creation. Once plan is created then Subscription is created by providing plan id.

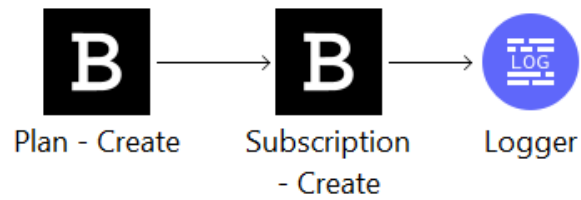
- In Anypoint Studio, click **File > New > Mule Project**, name the project and click Finish.
- In the search field, type **Http** and drag the **HTTP Listener** to the canvas.
- Click the **HTTP Listener**, click the green plus sign to the right of Connector Configuration. On the next screen, add the host and port, click OK
- Provide **Path** to HTTP Listener (Example: /recurring-billing)
- This mule application allows users to create a plan with or without trial period based on a flag `trialPeriodEnabled`.
- Add Set Variable and set variable name as `trialPeriodEnabled` and value as true if you want to add trial period during plan creation, false otherwise.
- This mule application allows users to do Create Subscription with payment method token or payment method nonce based on a flag `paymentMethodTokenEnabled`.
- Add Set Variable and set variable name as `paymentMethodTokenEnabled` and value as true if you want to use payment method token during subscription creation, false otherwise.
- If `trialPeriodEnabled` is not enabled and `paymentMethodTokenEnabled` is enabled then add FLOW Reference to call another sub flow 'create-plan-and-subscription-without-trial-period-using-nonce-detailsFlow' to create plan with trial period and create subscription using payment method token.
- If `trialPeriodEnabled` is not enabled and `paymentMethodTokenEnabled` is not enabled then add FLOW Reference to call another sub flow 'create-plan-and-subscription-without-trial-period-using-nonce-detailsFlow' to create plan with trial period and create subscription using payment method nonce.
- If `trialPeriodEnabled` is enabled and `paymentMethodTokenEnabled` is enabled then add FLOW Reference to call another sub flow 'create-plan-and-subscription-with-trial-period-token-detailsFlow' to create plan with trial period and create subscription using payment method nonce.
- If `trialPeriodEnabled` is enabled and `paymentMethodTokenEnabled` is not enabled then add FLOW Reference to call another sub flow 'create-plan-and-subscription-with-trial-period-nonce-detailsFlow' to create plan with trial period and create subscription using payment method nonce.
- Add Logger to log the response of the flow.

- On the Mule Palette, search for Braintree **Plan - Create** operation and drag it onto the canvas. Provide Plan, Trial Period and Billing details.
- Users cannot add AddOns or Discounts from API for Plan during Plan creation. Users need to link existing AddOns or Discounts to the Plan from the Braintree Account once the plan is created.
- If trialPeriodEnabled is enabled then the user needs to define trial period information in the Trial **Period Details** section during plan creation using **Plan - Create** operation.
- On the Mule Palette, search for Braintree **Subscription - Create** operation and drag it onto the canvas. Provide Subscription details with plan id and fields that user wants to override from plan in **General** tab.
- Set the Plan Id' to **payload.target.id** as expression in **Subscription - Create** operation
- In **Subscription - Create** operation, add AddOns details in **Add Ons** tab to add / update / remove add ons for subscription.
- In **Subscription - Create** operation, add descriptor information in **Descriptor** tab for subscription.
- In **Subscription - Create** operation, add Discount details in the **Discounts** tab to add / update / remove discounts for subscription.
- In **Subscription - Create** operation, subscription related options will be defined in the **Options** tab.
- If paymentMethodTokenEnabled is enabled then the user needs to select **Payment Method with token** option from **Payment Method** tab and provide token detail otherwise select **Payment Method with nonce** and provide nonce information in **Subscription - Create** operation.
- Drag logger onto canvas and log payload as expression.
- After creating the flow, right-click the project and click **Run As > Mule Application**.
- Once the application is deployed, use **HTTP Listener's Path** to execute the Mule Flow.

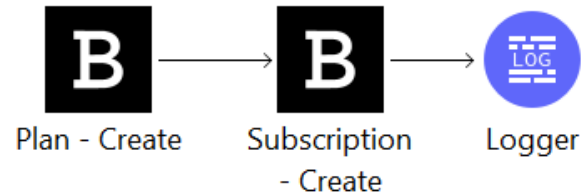
▼ create-plan-and-subscription-detailsFlow



▼ create-plan-and-subscription-without-trial-period-using-token-detailsFlow



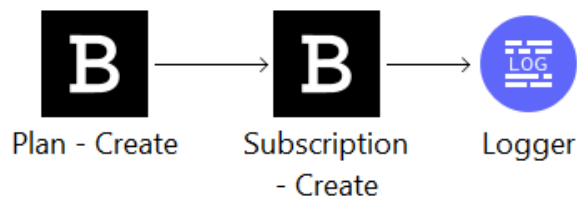
▼ create-plan-and-subscription-without-trial-period-using-nonce-detailsFlow



▼ create-plan-and-subscription-with-trial-period-nonce-detailsFlow



▼ create-plan-and-subscription-with-trial-period-token-detailsFlow



Use Case Configuration XML : Paste this XML code into Anypoint Studio to experiment with the flow described above.

```
<?xml version="1.0" encoding="UTF-8"?>
<mule
  xmlns:braintree="http://www.mulesoft.org/schema/mule/braintree"
  xmlns:ee="http://www.mulesoft.org/schema/mule/ee/core"
  xmlns:http="http://www.mulesoft.org/schema/mule/http"
  xmlns="http://www.mulesoft.org/schema/mule/core"
  xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.mulesoft.org/schema/mule/ee/core
    http://www.mulesoft.org/schema/mule/ee/core/current/mule-ee.xsd
```

```

http://www.mulesoft.org/schema/mule/core
http://www.mulesoft.org/schema/mule/core/current/mule.xsd
http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-
http.xsd
http://www.mulesoft.org/schema/mule/braintree
http://www.mulesoft.org/schema/mule/braintree/current/mule-braintree.xsd">
  <http:listener-config name="HTTP_Listener_config"
    doc:name="HTTP Listener config"
    doc:id="a3cfe521-b7b9-4a38-8048-ff1f02bf655d">
    <http:listener-connection host="0.0.0.0"
      port="8081" />
  </http:listener-config>
  <configuration-properties
    doc:name="Configuration properties"
    doc:id="5380985a-03a1-4a72-94dc-d32c69578389"
    file="mule-application.properties" />
  <braintree:config name="Braintree_Config"
    doc:name="Braintree Config"
    doc:id="358f0686-82fd-438d-93a6-e58a719c361e">
    <braintree:connection
      merchantId="{ Braintree.merchantId}"
      publicKey="{ Braintree.publicKey}"
      privateKey="{ Braintree.privateKey}"
      environment="{ Braintree.environment}" />
  </braintree:config>
  <flow name="create-plan-and-subscription-detailsFlow"
    doc:id="c97819d2-9e8c-4037-a385-3d40898df5ea">
    <http:listener doc:name="Listener"
      doc:id="5bf1b916-c6c5-43d3-870f-d09e76ec537f"
      config-ref="HTTP_Listener_config"
      path="{ Braintree.recurring.billing.demo.path}" />
    <set-variable value="{ Braintree.trial.period.enabled}"
      doc:name="Set Variable" doc:id="70b6b359-2b90-4a0a-bc8f-
f6ac04f960f9"
      variableName="trialPeriodEnabled" />
    <set-variable
      value="{ Braintree.recurring.billing.subscription.through.payment.token}"
      doc:name="Set Variable" doc:id="b47e6e0a-07e4-405a-aa6e-
29660740a479"
      variableName="paymentMethodTokenEnabled" />
    <choice doc:name="Choice"
      doc:id="2bf6329d-bc81-4dc3-a39f-217e87a09932">
      <when expression="#[vars.trialPeriodEnabled == 'false']">
        <choice doc:name="Choice"
          doc:id="8cd93e41-1038-4303-921c-4543ea7c7818">
          <when
expression="#[vars.paymentMethodTokenEnabled == 'false']">
            <flow-ref doc:name="Flow Reference"

```



```

a74c-040ab9b9cbc8" doc:id="6068f098-824a-4718-
subscription-without-trial-period-using-nonce-detailsFlow" />
name="create-plan-and-
</when>
<otherwise>
<flow-ref doc:name="Flow Reference"
doc:id="7265f181-5d16-41a4-
bc37-877e1f80a87a"
name="create-plan-and-
subscription-without-trial-period-using-token-detailsFlow" />
</otherwise>
</choice>
</when>
<otherwise>
<choice doc:name="Choice"
doc:id="d4707ea8-6725-4469-aaad-dffe2e931847">
<when
expression="#[vars.paymentMethodTokenEnabled == 'false']">
<flow-ref doc:name="Flow Reference"
doc:id="cfc27141-859f-47cb-
a348-2a08d2453809"
name="create-plan-and-
subscription-with-trial-period-nonce-detailsFlow" />
</when>
<otherwise>
<flow-ref doc:name="Flow Reference"
doc:id="6010fe42-6fac-4350-
a635-06389d0a8d86"
name="create-plan-and-subscription-with-trial-period-token-detailsFlow"
/>
</otherwise>
</choice>
</otherwise>
</choice>
</flow>
<sub-flow
name="create-plan-and-subscription-without-trial-period-using-token-detailsFlow"
doc:id="83bdafec-33ab-4484-aaca-8485fb7545c8">
<braintree:create-plan doc:name="Plan - Create"
doc:id="ff144eb7-7176-4596-88c4-140e8c13de26"
config-ref="Braintree_Config" name="TestPlan" price="0.04"
currencyIsoCode="USD" billingFrequency="1" billingDayOfMonth="12"
numberOfBillingCycles="15" />
<braintree:create-subscription
doc:name="Subscription - Create"
doc:id="d560cc59-abb5-41f3-988e-52916998199e"
config-ref="Braintree_Config" planId="#[payload.target.id]"
neverExpires="true">
<braintree:subscription-payment-mode>
<braintree:payment-method-with-token

```

```

subscriptionPaymentMethodToken="{ Braintree.recurring.billing.subscription.payment.token}" />
    </braintree:subscription-payment-mode>
</braintree:create-subscription>
<logger level="INFO" doc:name="Logger"
    doc:id="45aff83f-4aa9-4cc5-bfae-9b7511093019" message="#[payload]"
/>
</sub-flow>

<sub-flow
    name="create-plan-and-subscription-without-trial-period-using-nonce-detailsFlow"
    doc:id="1414d1c3-7669-485e-a61c-7a13ef7cce17">
    <braintree:create-plan doc:name="Plan - Create"
        doc:id="898883cc-e820-4f65-9e04-08244ed25087"
        config-ref="Braintree_Config" name="TestPlan" price="0.04"
        currencyIsoCode="USD" billingFrequency="1" billingDayOfMonth="12"
        numberOfBillingCycles="15" />
    <braintree:create-subscription
        doc:name="Subscription - Create"
        doc:id="3723d613-f0b8-4755-8fad-a8e9e24c3dfc"
        config-ref="Braintree_Config" planId="#[payload.target.id]"
        neverExpires="true">
        <braintree:subscription-payment-mode>
            <braintree:payment-method-with-nonce
subscriptionPaymentMethodNonce="{ Braintree.recurring.billing.subscription.payment.nonce}" />
            </braintree:subscription-payment-mode>
        </braintree:create-subscription>
        <logger level="INFO" doc:name="Logger"
            doc:id="610cc9a1-e73c-4501-ba8c-eb74cac90912"
message="#[payload]" />
        </sub-flow>

        <sub-flow
            name="create-plan-and-subscription-with-trial-period-nonce-detailsFlow"
            doc:id="9fe63fc6-42c6-4d2c-9952-6a85a8fb9617">
            <braintree:create-plan doc:name="Plan - Create"
                doc:id="f7000339-c7b2-4fff-968f-686c11331f8f"
                config-ref="Braintree_Config" name="TestPlanWithTrialPeriod"
                price="0.04" currencyIsoCode="USD" billingFrequency="1"
                numberOfBillingCycles="15">
                <braintree:trial-period trialDuration="1"
                    trialDurationUnit="DAY" />
            </braintree:create-plan>
            <braintree:create-subscription
                doc:name="Subscription - Create"
                doc:id="85209221-b8f0-49bc-a8cc-3c9292fd8840"
                config-ref="Braintree_Config" planId="#[payload.target.id]"
                trialPeriod="true" trialDuration="1" trialDurationUnit="DAY"
                neverExpires="true">
                <braintree:subscription-payment-mode>
                    <braintree:payment-method-with-nonce
subscriptionPaymentMethodNonce="{ Braintree.recurring.billing.subscription.payment.nonce}" />
                    </braintree:subscription-payment-mode>
            </braintree:create-subscription>
        </sub-flow>
    </braintree:create-subscription>
</sub-flow>

```

```

        </braintree:create-subscription>
        <logger level="INFO" doc:name="Logger"
            doc:id="9c631969-54fa-4fe2-a9f4-f4e9fb67b33a" message="#[payload]"
/>
    </sub-flow>
    <sub-flow
        name="create-plan-and-subscription-with-trial-period-token-detailsFlow"
        doc:id="eb6e36bf-1b16-43be-b5d9-672f1f31881f">
        <braintree:create-plan doc:name="Plan - Create"
            doc:id="63e6d956-d1a4-414d-957c-bf3d58d3d4b4"
            config-ref="Braintree_Config" name="TestPlanWithTrialPeriod"
            price="0.04" currencyIsoCode="USD" billingFrequency="1"
            numberOfBillingCycles="15">
            <braintree:trial-period trialDuration="1"
                trialDurationUnit="DAY" />
        </braintree:create-plan>
        <braintree:create-subscription
            doc:name="Subscription - Create"
            doc:id="88d14a3b-4bcd-4b98-91e0-f09ea5cc724a"
            config-ref="Braintree_Config" planId="#[payload.target.id]"
            trialPeriod="true" trialDuration="1" trialDurationUnit="DAY"
            neverExpires="true">
            <braintree:subscription-payment-mode>
                <braintree:payment-method-with-token
                    subscriptionPaymentMethodToken="{ Braintree.recurring.billing.subscription.payment.token
}" />
            </braintree:subscription-payment-mode>
        </braintree:create-subscription>
        <logger level="INFO" doc:name="Logger"
            doc:id="4dc6bf49-5d94-470e-98b1-dc5cd3dd6929"
            message="#[payload]" />
    </sub-flow>
</mule>

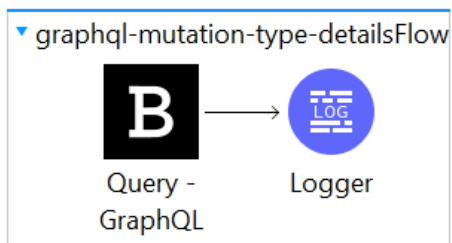
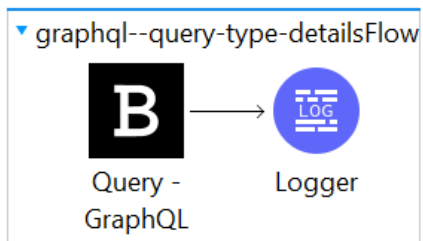
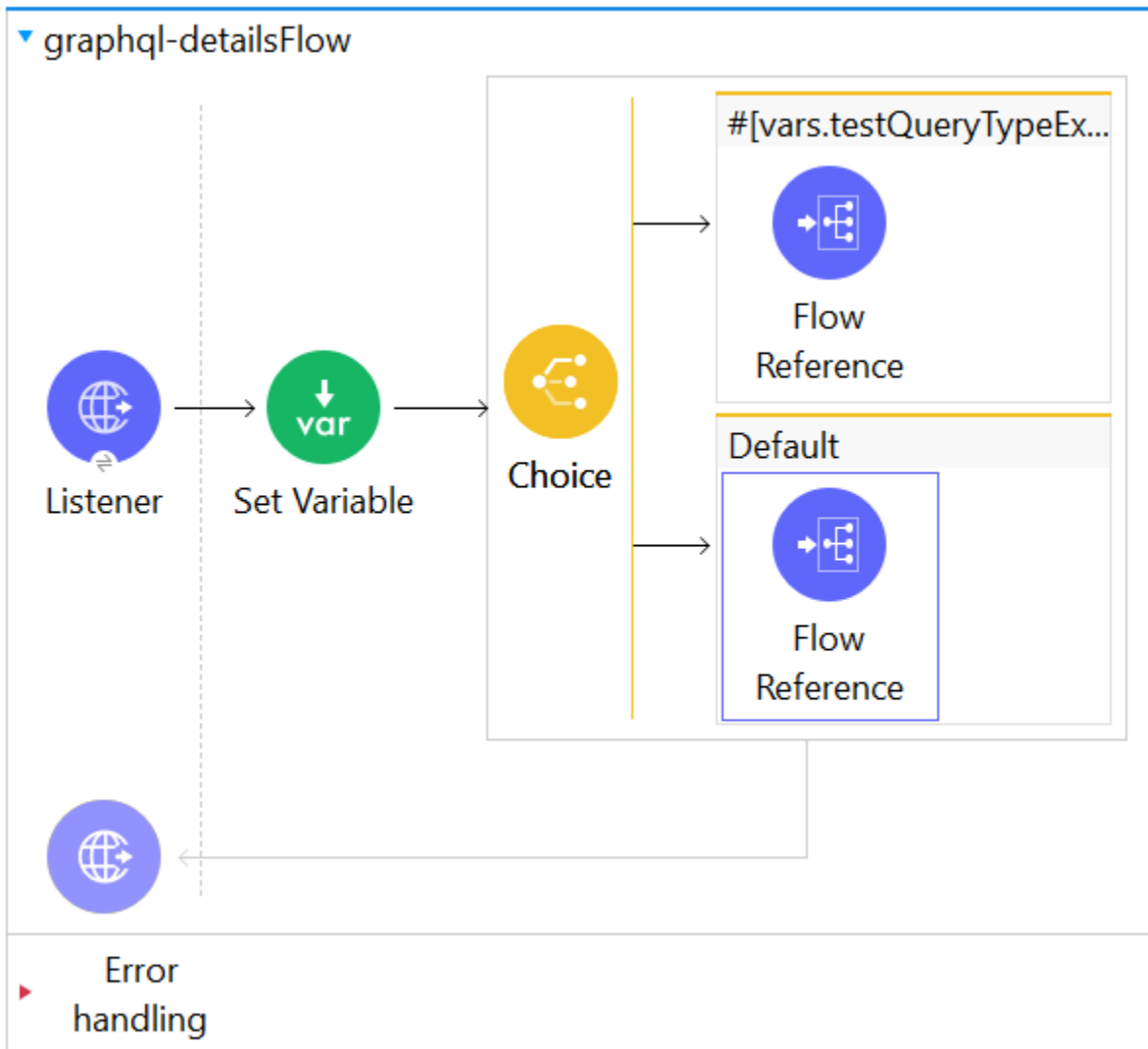
```

GraphQL Example for Query and Mutation

This use case demonstrates usage of GraphQL operation for both Query as well as Mutation root element in GraphQL. Queries are used to fetch data. Mutations are used to make requests that create or modify data.

- In Anypoint Studio, click **File > New > Mule Project**, name the project and click Finish.
- In the search field, type **Http** and drag the **HTTP Listener** to the canvas.
- Click the **HTTP Listener**, click the green plus sign to the right of Connector Configuration. On the next screen, add the host and port, click OK

- Provide **Path** to HTTP Listener (Example: /graphql)
- This mule application allows users to use query type or mutation type in GraphQL and get response in expected format defined in query definition using GraphQL with provided variables and definition.
- Add Set Variable and set variable name as testQueryTypeExampleEnabled and value as true if you want to use a query type element to fetch data. If users want to use mutation type then need to set it to false.
- If testQueryTypeExampleEnabled is enabled then add FLOW Reference to call another sub flow 'graphql--query-type-detailsFlow' to fetch data by using query type in query definition.
- If testQueryTypeExampleEnabled is not enabled then add FLOW Reference to call another sub flow 'graphql-mutation-type-detailsFlow' to create / modify data..
- Add Logger to log the response of the flow.
- On the Mule Palette, search for Braintree **Query - GraphQL** operation and drag it onto the canvas. Provide Query Definition and Query Variables if any.
- Drag logger onto canvas and log payload as expression.
- After creating the flow, right-click the project and click **Run As > Mule Application**.
- Once the application is deployed, use **HTTP Listener's Path** to execute the Mule Flow.



Use Case Configuration XML : Paste this XML code into Anypoint Studio to experiment with the flow described above.

```

<?xml version="1.0" encoding="UTF-8"?>

<mule
  xmlns:braintree="http://www.mulesoft.org/schema/mule/braintree"
  xmlns:ee="http://www.mulesoft.org/schema/mule/ee/core"
  xmlns:http="http://www.mulesoft.org/schema/mule/http"
  xmlns="http://www.mulesoft.org/schema/mule/core"
  xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
http://www.mulesoft.org/schema/mule/ee/core
http://www.mulesoft.org/schema/mule/ee/core/current/mule-ee.xsd
http://www.mulesoft.org/schema/mule/core
http://www.mulesoft.org/schema/mule/core/current/mule.xsd
http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-
http.xsd
http://www.mulesoft.org/schema/mule/braintree
http://www.mulesoft.org/schema/mule/braintree/current/mule-braintree.xsd">
  <http:listener-config name="HTTP_Listener_config"
    doc:name="HTTP Listener config"
    doc:id="a3cfe521-b7b9-4a38-8048-ff1f02bf655d">
    <http:listener-connection host="0.0.0.0"
      port="8081" />
  </http:listener-config>
  <configuration-properties
    doc:name="Configuration properties"
    doc:id="5380985a-03a1-4a72-94dc-d32c69578389"
    file="mule-application.properties" />
  <braintree:config name="Braintree_Config"
    doc:name="Braintree Config"
    doc:id="358f0686-82fd-438d-93a6-e58a719c361e">
    <braintree:connection
      merchantId="{ Braintree.merchantId}"
      publicKey="{ Braintree.publicKey}"
      privateKey="{ Braintree.privateKey}"
      environment="{ Braintree.environment}" />
  </braintree:config>
  <flow name="graphql-detailsFlow"
    doc:id="ffa17cb3-4649-415c-9ebb-a52bef5e19b0">
    <http:listener doc:name="Listener"
      doc:id="eee4dce6-a28c-499d-b52b-aa8747e39e66"
      config-ref="HTTP_Listener_config"
      path="{ Braintree.graphql.example.path}" />
    <set-variable
      value="{ Braintree.graphql.test.query.type.example}"
      doc:name="Set Variable" doc:id="7e0d9236-f9f3-48a9-9b0a-
a00274f5a549"
      variableName="testQueryTypeExampleEnabled" />

    <choice doc:name="Choice"
      doc:id="8cfd208a-cd73-4a1e-8838-f0897c9dffe">

```

```

        <when expression="#[vars.testQueryTypeExampleEnabled == 'true']">
            <flow-ref doc:name="Flow Reference"
                doc:id="77375bb0-834f-4dd0-920c-a687f7aa76ac"
                name="graphql--query-type-detailsFlow" />
        </when>
        <otherwise>
            <flow-ref doc:name="Flow Reference"
                doc:id="d44350b6-5ad2-4a33-b530-4383b581e805"
                name="graphql-mutation-type-detailsFlow" />
        </otherwise>
    </choice>
</flow>
<sub-flow name="graphql--query-type-detailsFlow"
    doc:id="c8a37f2e-c2f6-456e-9502-79c6be8263ff">
    <brantree:query-graph-ql
        doc:name="Query - GraphQL"
        doc:id="956c2da4-aac8-45cf-9a3d-53ea15da9083"
        config-ref="Braintree_Config">
        <brantree:query-
definition><![CDATA[#{ Braintree.graphql.query.type.example.query.definition}]]></brantree:query-
definition>

        <brantree:variables><![CDATA[#{ Braintree.graphql.query.type.example.query.variable}]]><
/brantree:variables>
        </brantree:query-graph-ql>
        <logger level="INFO" doc:name="Logger"
            doc:id="4e71358d-f1de-4f9c-8f78-858b845865e4" message="#[payload]"
/>
    </sub-flow>
    <sub-flow name="graphql-mutation-type-detailsFlow"
        doc:id="37a28bd1-0666-4aff-b4e8-10c62b9237e9">
        <brantree:query-graph-ql
            doc:name="Query - GraphQL"
            doc:id="1e7ef861-0a18-4830-8529-29c0e2f81493"
            config-ref="Braintree_Config">
            <brantree:query-
definition><![CDATA[#{ Braintree.graphql.mutation.type.example.query.definition}]]></brantree:que
ry-definition>

            <brantree:variables><![CDATA[#{ Braintree.graphql.mutation.type.example.query.variable}]]
></brantree:variables>
            </brantree:query-graph-ql>
            <logger level="INFO" doc:name="Logger"
                doc:id="965c1210-6f7e-4c3d-b34d-1f85d1c3a8fa" message="#[payload]"
/>
        </sub-flow>
</mule>

```

Resources

- [Braintree Connector Release Note.](#)
- [Braintree Connector API docs.](#)