

Braintree Connector

B

Table of Contents

Overview

Introduction

Software Requirements

Installation

Connector Namespace and Schema

Maven Dependency Information

Configuration

- Creating a New Project

- Configuring the Braintree Global Element

- Understanding the Braintree Connector

Common Use Cases

- Create a Customer with Payment Method and View Details

- Undo a transaction

- Submit transaction for settlement

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 provide 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

TIP

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 pom.xml file in the Mule project.

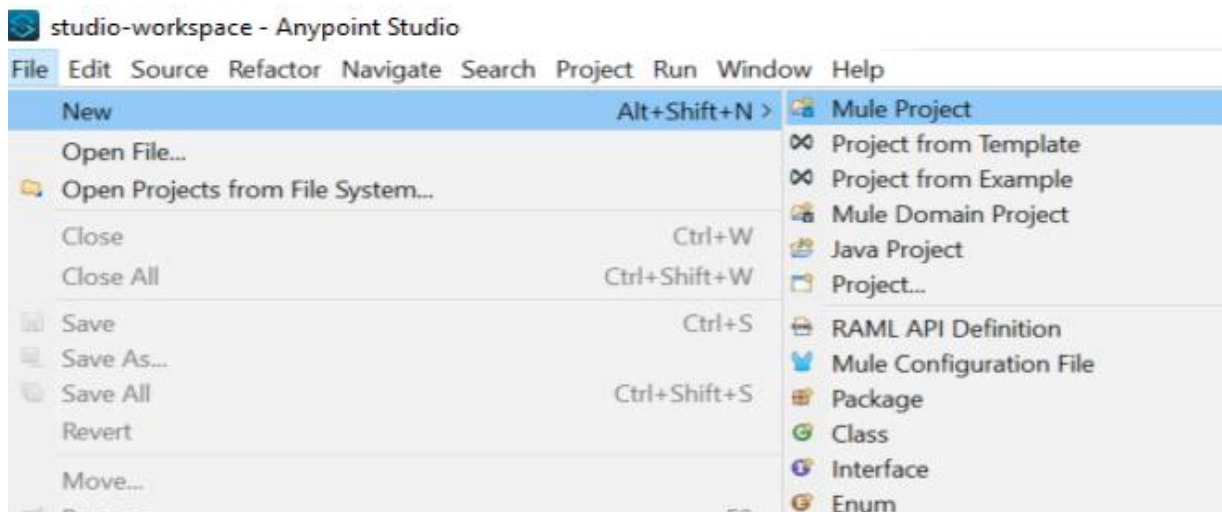
```
<dependency>
  <groupId>com.mulesoft.connectors</groupId>
  <artifactId>mule-braintree-connector</artifactId>
  <version>1.0.0</version>
  <classifier>mule-plugin</classifier>
</dependency>
```

Configuration

Creating a New Project

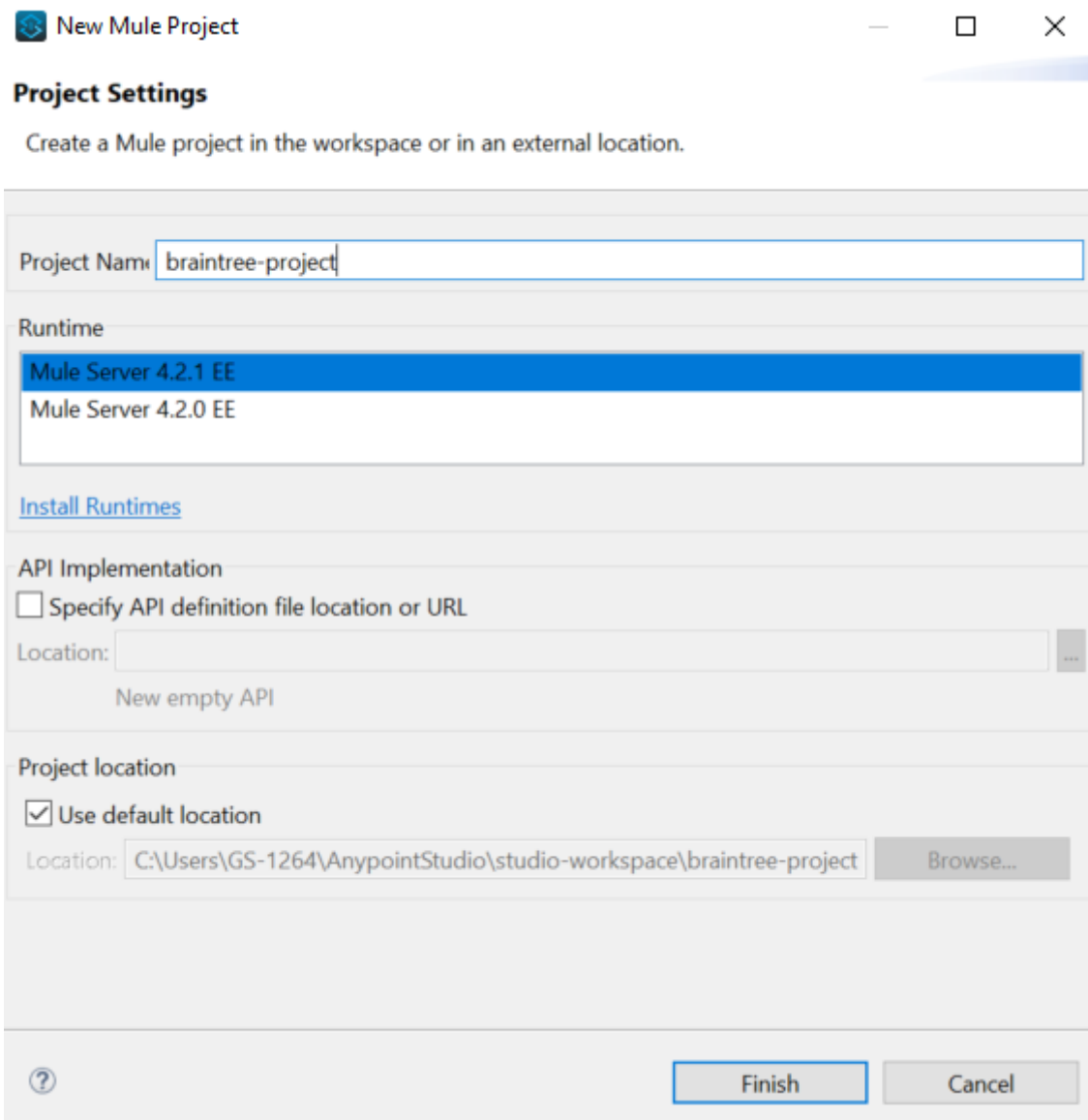
To use the Braintree connector in a Mule application project:

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, user must configure a global Braintree element that is used by the Braintree connector to authenticate. The Braintree connector supports API credentials authentication.

Authentication

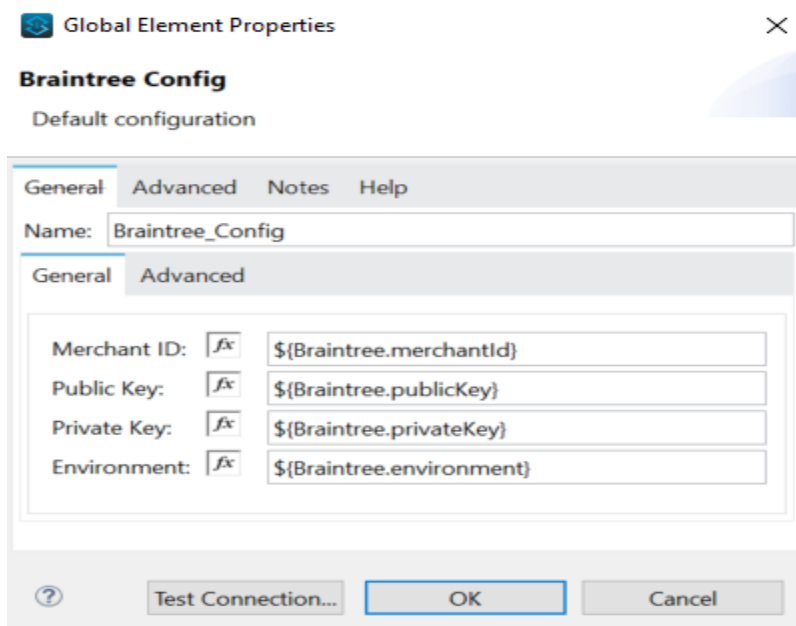
To access Braintree server, user needs to configure API credentials as

mentioned below.

API credentials are unique account identifiers that must be used to configure 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 user-specific public identifier. Each user associated with your Braintree gateway will have their own public key.
Private key	This is 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.



Understanding the Braintree Connector

The Braintree connector functions within a Mule application. Braintree connector allows you to perform Braintree server operations through mule application. 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 Braintree server. Braintree Connector is built using braintree-java Java API client provided by Braintree. It executes operations exposed by Braintree using configured parameters.

Braintree connector is configured to authorize access to Braintree server resources. It requires Public key, Private key, and Merchant ID 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 let's 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 developer can refer to [Braintree Connector API docs](#) and [Braintree Documentation](#) for more information about complex input or return types.

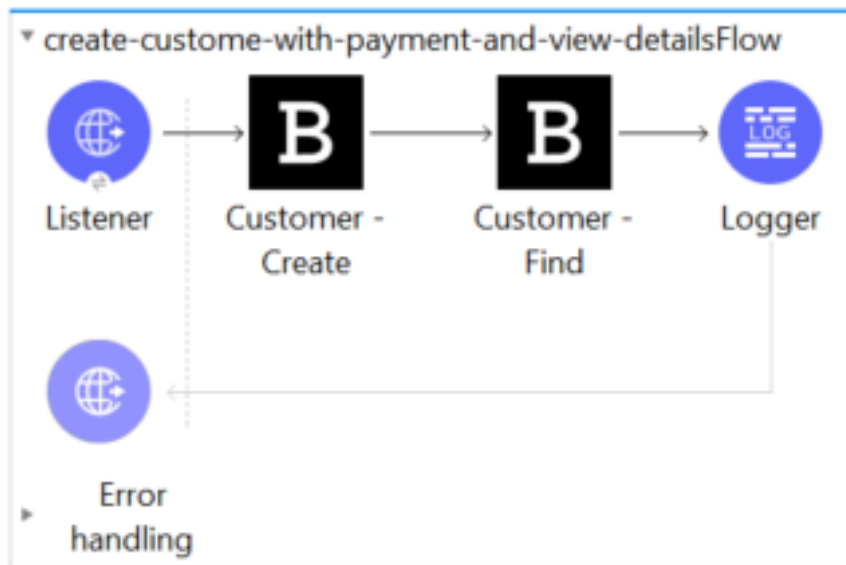
Common Use Cases

- Create a Customer with Payment Method and View Details
- Undo a transaction
- Submit transaction for settlement.

Create a Customer with Payment Method and View Details

This use case demonstrates the creating 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 **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 application is deployed, use **HTTP Listener's Path** to execute the MuleFlow.



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 />
</flow>
</configuration>
```

```

        <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

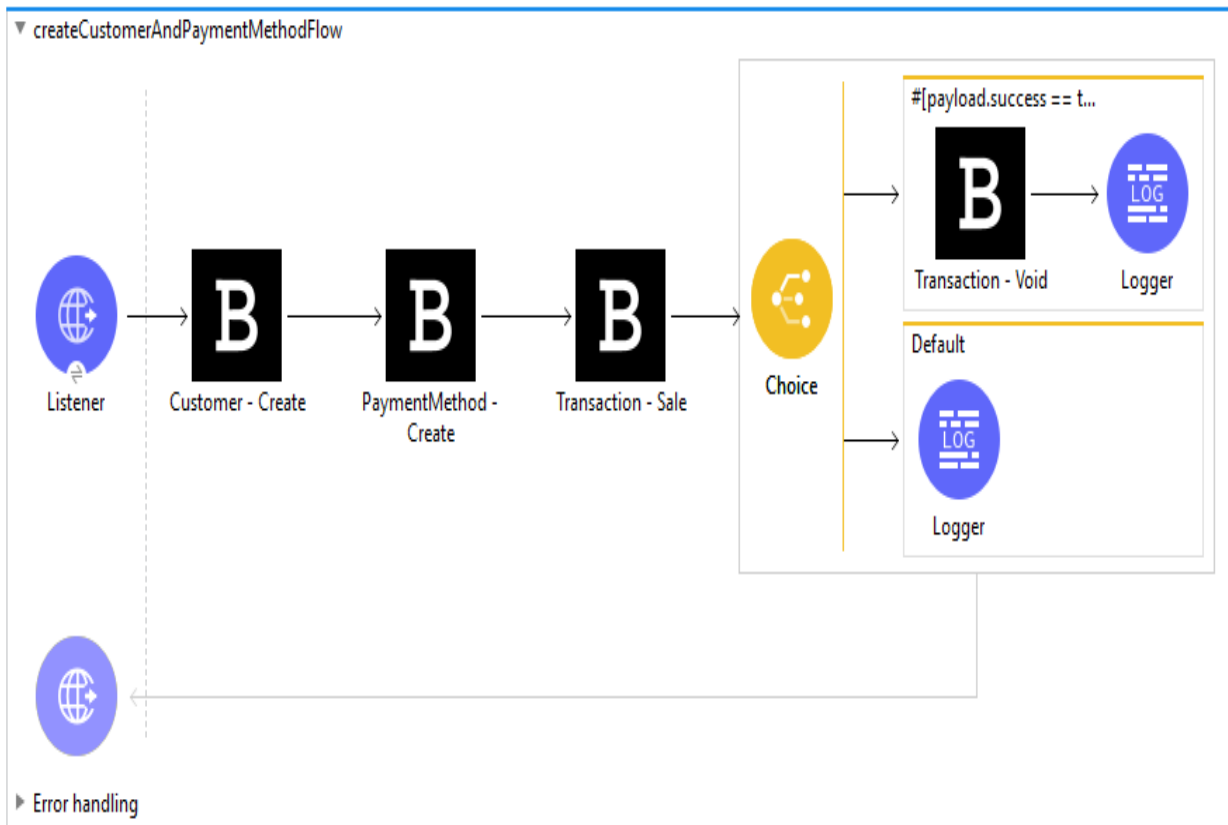
This use case demonstrates creating a Customer and Payment Method, perform Transaction then void the transaction before it settles.

- 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 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>
<brantree:config name="Braintree_Config" doc:name="Braintree Config"
    doc:id="d3788405-ab3f-4c9c-9de0-3fdb64f261ec">
    <brantree: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" />
    <brantree: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" />
    <brantree: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}" />
</flow>
```

```

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

<choice doc:name="Choice" doc:id="9522224f-3cb0-4d72-aedd-26976684a68b">
    <when expression="#[payload.success == true]">
        <brantree: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 was 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 user 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 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 provided expression.

- Add below condition as expression in **Choice** to check if sale transaction operation is a success.

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

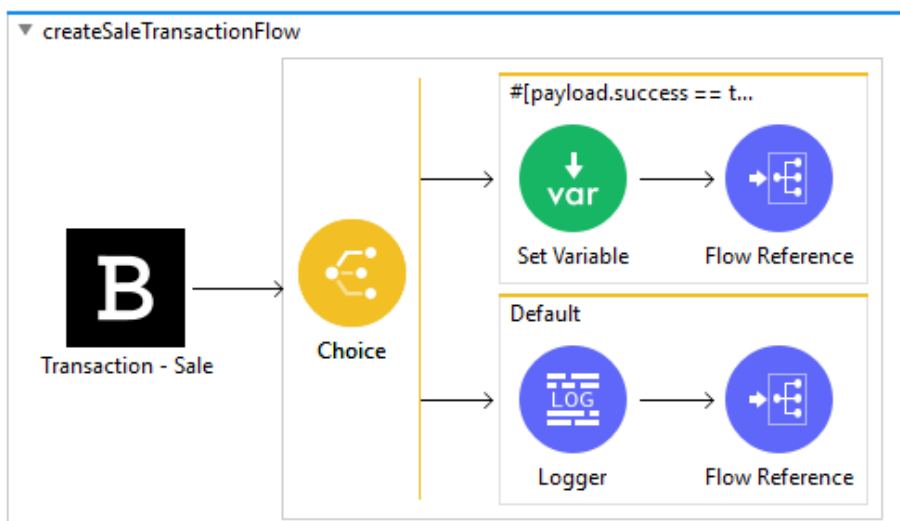
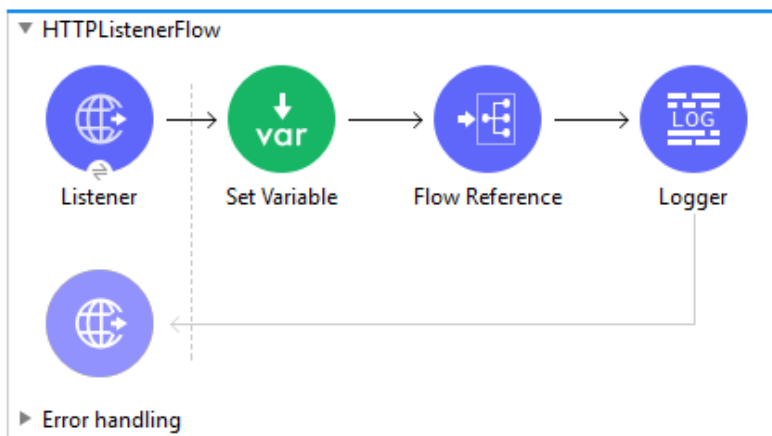
- To the expression side, add **Set Variable** with variable name as **transactionID** and value as expression **payload.target.id**. This transaction ID is later used by **Transaction - Find** to get the status of the transaction.
- Add **Flow Reference** to call another sub flow '**submitTransactionForSettlementFlow**' to submit transaction 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 same sub flow '**createSaleTransactionFlow**' a gain.
- Create new sub flow **submitTransactionForSettlementFlow**.
- It submit transaction 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 provided expression.
- Add below condition as expression in **Choice** to check if transaction is AUTHORIZED and you want to do full settlement.

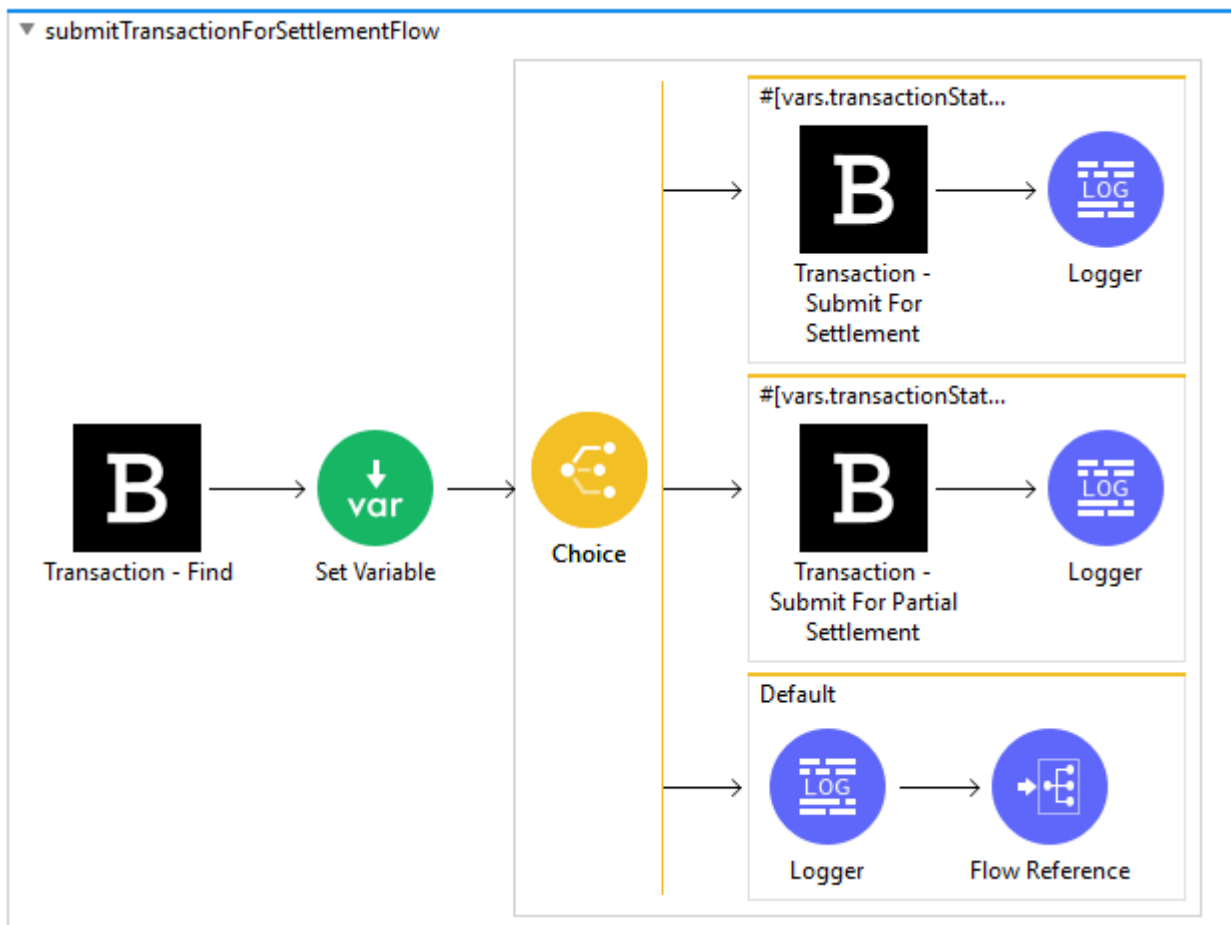
```
vars.transactionStatus == 'AUTHORIZED' and vars.isPartialSettlement == 'false'
```

- 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 same sub flow '**submitTransactionForSettlementFlow**' again.
- After creating the flow, right-click the project and click **Run As > Mule Application**.
- Once 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>

<braintree:config name="Braintree_Config" doc:name="Braintree Config"
  doc:id="51a76243-e2a9-4b30-ac0e-1b0fe91720cd">
  <braintree: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">

    <braintree:sale doc:name="Transaction - Sale"

        doc:id="108a09d1-ec8f-4195-a090-b1497bd3a692"
        saleAmount="${transaction.sale.amount}"

        config-ref="Braintree_Config">

            <braintree:sale-customer>

                <braintree:new-customer customerFirstName="${customer.firstName}"

                    customerLastName="${customer.lastName}" email="${customer.email}"

                    phone="${customer.phone}" />

            </braintree:sale-customer>

            <braintree:sale-payment-options>

                <braintree:card-details cardholderName="${paymentMethod.cardholderName}"

                    number="${paymentMethod.number}" cvv="${paymentMethod.cvv}"

                    expirationMonth="${paymentMethod.expirationMonth}"
                    expirationYear="${paymentMethod.expirationYear}"/>

            </braintree:sale-payment-options>

            <braintree:sale-options />

            <braintree: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>

        </choice>

    </sub-flow>
```

```
</when>

<otherwise>

    <logger level="INFO" doc:name="Logger"

        doc:id="8ddcd4b1-d005-48fd-9a38-27087416a568"
message='#["Transaction was 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">

    <braintree: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']">

                <braintree: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']">

    <braintree:submit-for-partial-settlement

        doc:name="Transaction - Submit For Partial Settlement"
doc:id="28e9cbc1-2ddf-4d9a-abc9-79ece0a9f5d1"

        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>
```

Resources

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