Authorize.NET Connector

Overview

Authorize.Net connector is an Integration to payment gateway which enables users to automatically charge their customers and accepts ACH and Credit card payments.It is a visual integration to have Credit card or ACH payment received from the UI. This visual component triggers an integration. It is used for storing the connection properties which we will leverage during a transaction triggered from Salesforce.

Process Summary

- The user is provided with a simple API that needs to be written along with a simple HTML with payment details.
- A predefined JSON is provided where the user will attach credit card details to accept the form which is automatically rendered
- We are providing a Javascript library that displays the information visually on the screen
- The payment is posted through an API and the response is posted to the user.
- The processing of the response can be posted to Salesforce, Quickbooks or any browser based application.
- When a payment is made the connector triggers the integration into the system that they are using for instance Salesforce, quickbook etc or any browser based application.
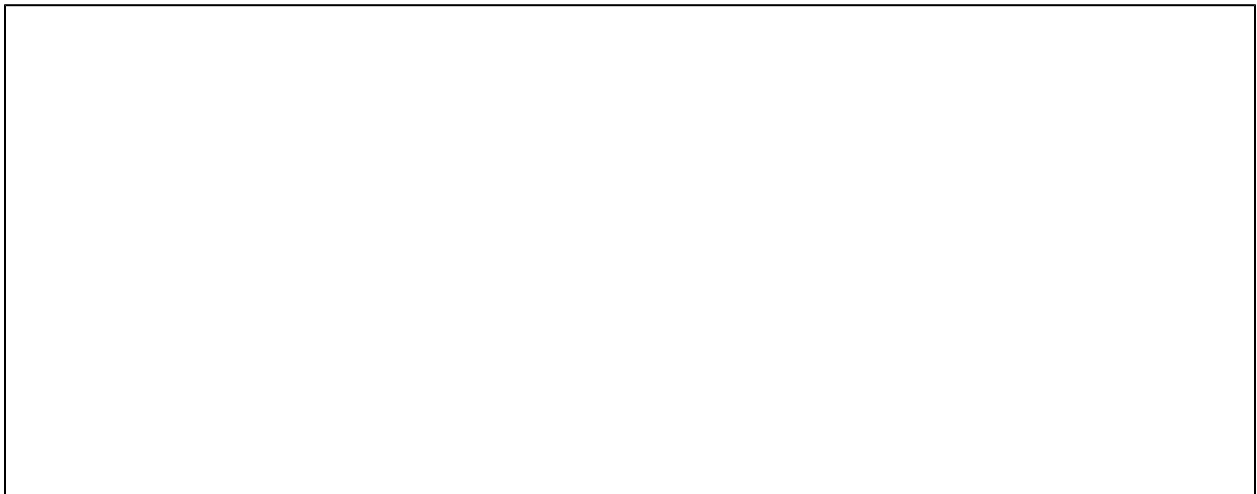
Steps to Follow

Requirements:

1. Connect Salesforce with Authorize.net for payment processing gateway
2. Compatibility to connect with any Salesforce custom object or standard object to process payments from
3. Save card details within Salesforce with PCI compliance
4. Login to DBSync Workflow Instance
5. Click on Development Studio
6. Salesforce setup-

   Configure Authorize.Net DBSyn package.

   Verify, click and process payment in visual force page of Salesforce Instance

1. Create a workflow with Console Adapter as reader & Salesforce as writer.
2. Create and setup Authorize.net connector & save.
3. Use the predefined sample JSON in Advance Query Builder as given below.
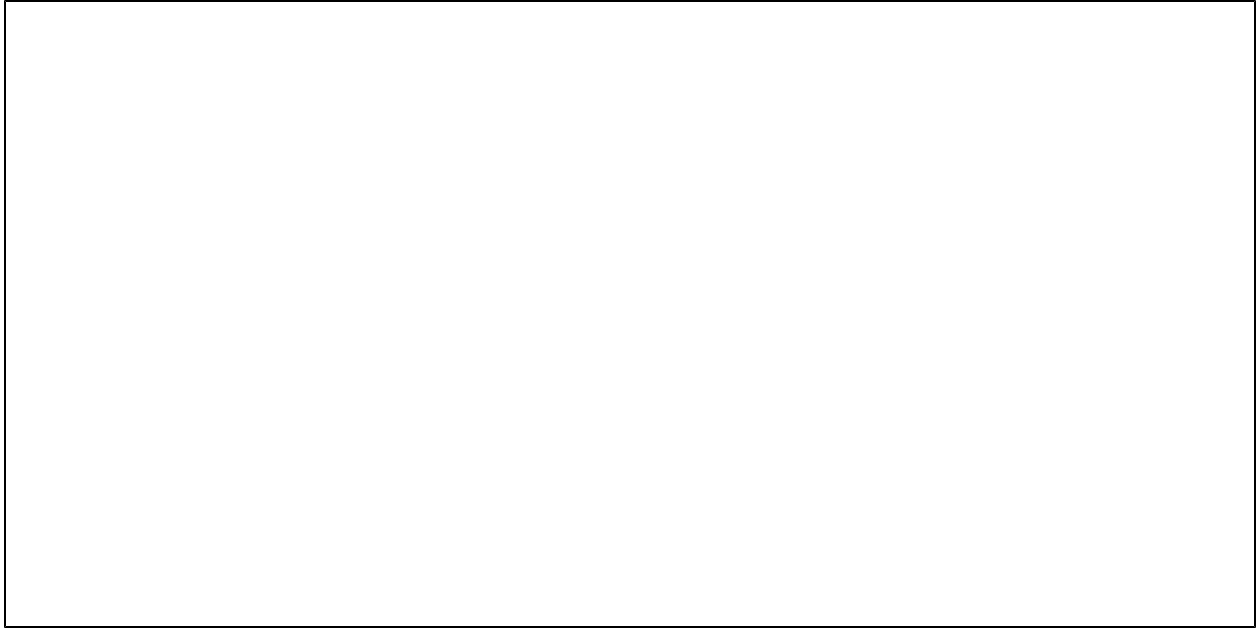
Fig : Advance Query

To use ACH for transaction instead of Credit Card, the predefined sample JSON for query builder in workflow & apex code for Visualforce Page is given below.

There are 3 sections of JSON query necessary while setup - Call, Request , Response

Section 1 :

This section is internally created by DBSync and is specifically designed to run an API call as mentioned below. A project and respective processes are triggered asynchronously.

```
{
  "call": {
  "process": "process-xxxx",
  "session": [
  {
  "value": "xxxxxx",
  "key": "xxxxxx"
  },
  {
  "value": "xxxxxx",
  "key": "xxxx"
  }
  ],
  "project": "project-xxxx",
  "run": "async"
 },
```

Section 2:

Payload request is coded on the basis of Authorize.Net specification. Payload is generated based on user requirements. For example, Bill to, amount, translation type, and Credit card informationare the required fields displayed for the below example.

```
  "request": {
  "transactionType": "authCaptureTransaction",
```

"amount": "1",

"billTo": {

"zip": "xxxxxx",

"firstName": "xxxxx",

"lastName": "xxxxxx",

"country": "XXX",

"address": "xxxxx",

"city": "xxxxxxx",

"state": "xx"

},

"payment": {

"creditCard": {

"cardCode": "XXX",

"cardNumber": "000XXXXXXX",

"expirationDate": "0000-XX"

}

}

Section 3:

Based on the information sent by Authorize.Net there is a response automatically generated which has all the information from the transaction.

This information can come from connected respective application representing the transaction that is taking place

},

"response": {

"transactionResponse": {

"cvvResultCode": "M",

"SupplementalDataQualificationIndicator": 0,

"transHashSha2": "",

"authCode": "EPDAYO",

"cavvResultCode": "2",

"transId": "60163622801",

"transHash": "",

"accountType": "Visa",

"accountNumber": "XXXX1111",

"networkTransId": "43UZUTCVIGA9D8HRVZ5JWQT",

"responseCode": "1",

"avsResultCode": "A",

"testRequest": "0",

"messages": [

{

"code": "1",

"description": "This transaction has been approved."

```
}
],
"refTransID": ""
},
"messages": {
"resultCode": "Ok",
"message": [
{
"code": "I00001",
"text": "Successful."
}
]
}
}
}
```

- Once the request comes in there is a Session_put.
- During the run of this call, you can out some things that persist across this run time which called session variables. Enter these values and make it static for the function to run

- This predefined JSON helps in getting the schema for mapping.
- In mapping, use the desired fields in the schema to update to Salesforce.

Mapping Screen Configuration

In this example, the encryption type is the Card number, and map the required values based on the target system.
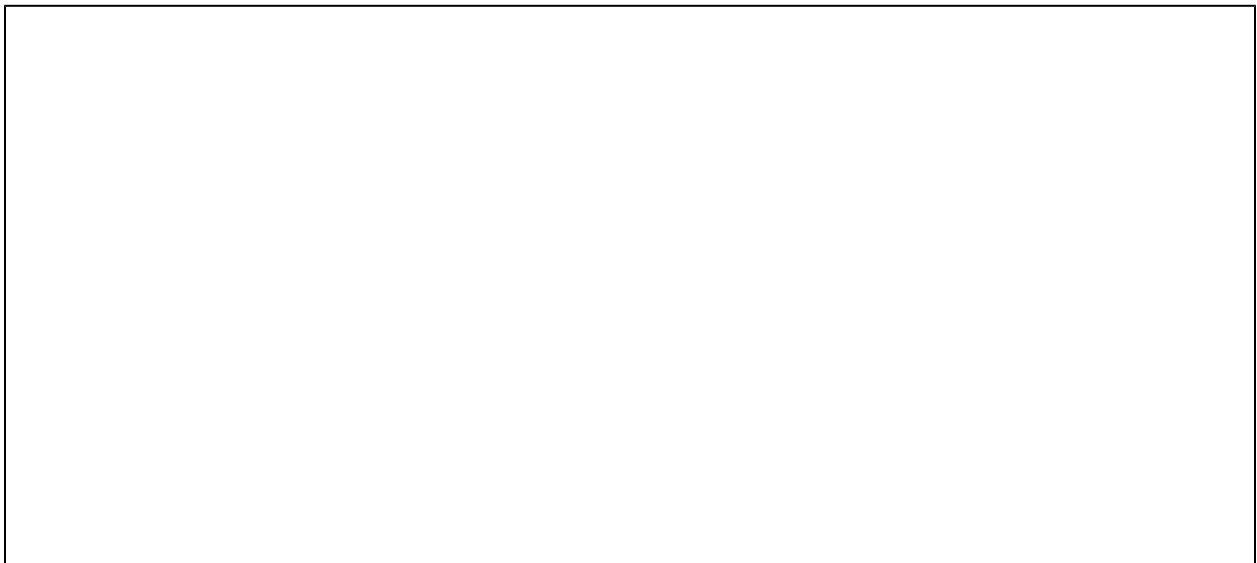

Fig : Mapping Screen

- Login to Salesforce with the same account used in the Cloud Workflow.
- Create a Visualforce Page
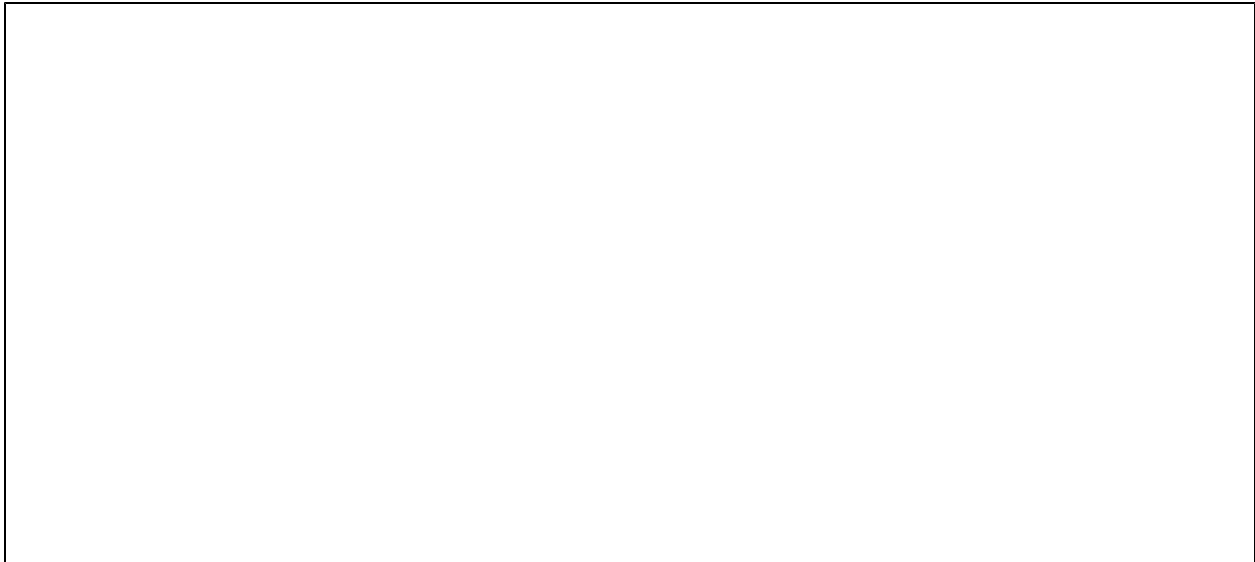- The visualforce page is an editable JSON which appear as below:

Fig : Salesforce Visualforce Page

- Click Edit use the same apex code given below to create the Visualforce page

```
<apex:page standardController="Opportunity" showHeader="false" standardStylesheets="false">
  <head lang="en">
   <meta charset="utf-8"/>
   <meta name="viewport" content="width=device-width,initial-scale=1.0"/>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://qa.mydbsync.com/cw/scripts/dbsync-cc.js"></script>
  </head>
  <body>
    <div id="cc"></div>
  </body>
  <apex:iframe src="" scrolling="true" id="cardFrame"/>
  <form id="formCardFrame" method="post"
      action="https://qa.mydbsync.com/cw/c/pages/card_payment" style="display:none;">
    <input name="json" value="" id="json" />
    <input type="submit"/>
  </form>
<script>
  $(document).ready(() => {
    const jsonRequest = {
    baseURL: "https://qa.mydbsync.com/DBSync",
      dbsyncAuth: {
        key: "xxxxx@xxxx.com",
```

Note:

Provide the user name and password for the DBsync Cloud Workflow instance in Key and Token placeholders respectively.

```
        token: "xxxxxx"
      },
```

```
          salesforce_session: "23234324243",

          paymentInstance: "AuthorizeNet",

          transactionRequest: {

            amount: "1",

            payment: {

              creditCard: {

                cardNumber: "5424000000000015",

                expirationDate: "2021-12",

                cardCode: "900"

              }

            },

            billTo: {

              firstName: "Ellen",

              lastName: "Johnson",

              company: "Souveniropolis",

              address: "14 Main Street",

              city: "Pecan Springs",

              state: "TX",

              zip: "46201",

              country: "USA"

            }

          },

          call: {

            project:"AuthorizeNetPayment",

            process:"ProcessAuthNetPayment_Copy",

            session: [

                  {"key":"opportunityId","value":"{!Opportunity.Id}"},

                  {"key":"object","value":"Opportunity"},

            ]

          }

        };

              // $('#cc').acceptCC(jsonRequest);


      $('#json').val(JSON.stringify(jsonRequest));

      $('#formCardFrame').submit();

    })

</script>

</apex:page>
```

Note:

- The credit card information is not saved unless chosen as an option and as default these values are empty. The user decides which CC

information must be stored and whether if/not to be stored. If values of the CC are provided by the user it will appear in this section if not it is set as empty and needs to be manually entered and this decision is up to to the user.

- Click Save

- Make sure to check dbsyncAuth, project, process, payment instance fields are relative to your DBSync account.
- Also, make sure you map fields of transaction request with the required Salesforce fields of your choice.
- Map the session fields for the fields which are to be used in CloudWorkflow Mapping.
- With this Visualforce Page, create a Lightning Action for the required Salesforce Object.
- Also, make sure that you add the Lightning Action to the same Salesforce Object's page layout.
- Now when you click on the Lightning Action button for a record, there will be a pop-up as shown below.

Now the setup is ready.

Salesforce and ACH Final setup view

- Click on the Account you wish to process the payment
- Click the drop-down under the account name to select the User to process the payment

- From the list select the Visual Force page you created.
- There will be a pop-up as shown below.

Fig : Auto-filled Display of CC details

- Here, the fields will be auto-filled with the values you mapped against the fields in the Visualforce Page.
- You can make changes if required.
- On click of 'Make a payment!', you will see either a success or failure response as shown below
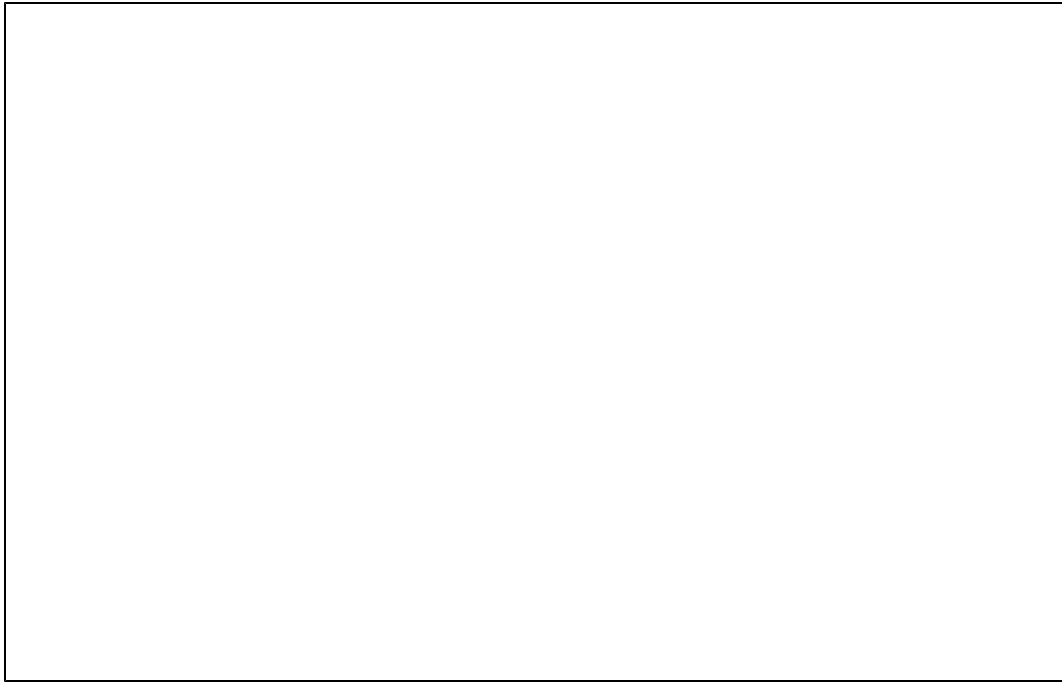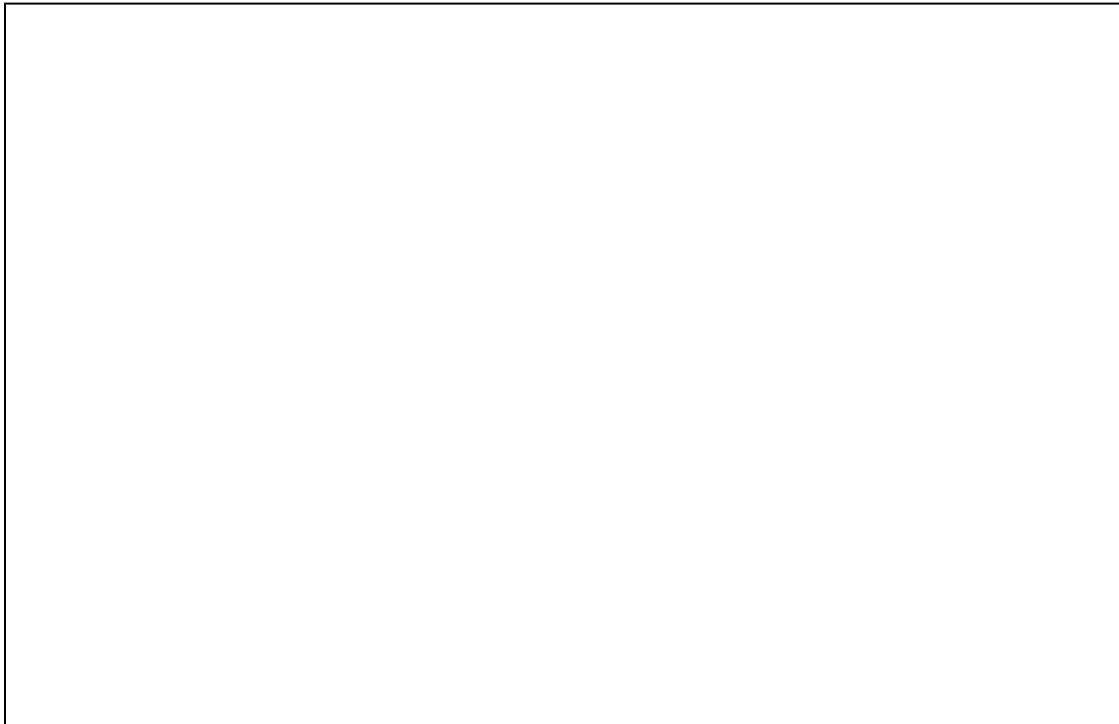
Fig : Transaction Successful

Fig : Transaction Failed

- Only on success, your workflow defined in your DBSync account will execute & update the required fields in Salesforce for the fields that you have mapped in your DBSync account.
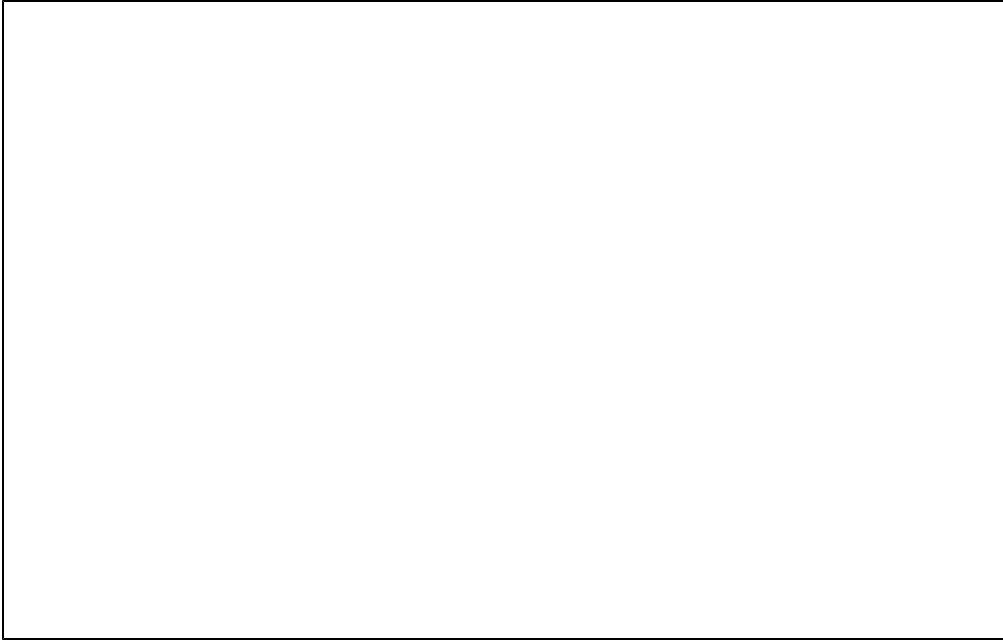- The pop-up for ACH looks like this.

Fig : ACH Visual View