TGH

Making Integrations Simpler

boomi Partner

Implementing Manual OAuth 2.0 with Client Credentials

Author

Asish Palatasingh

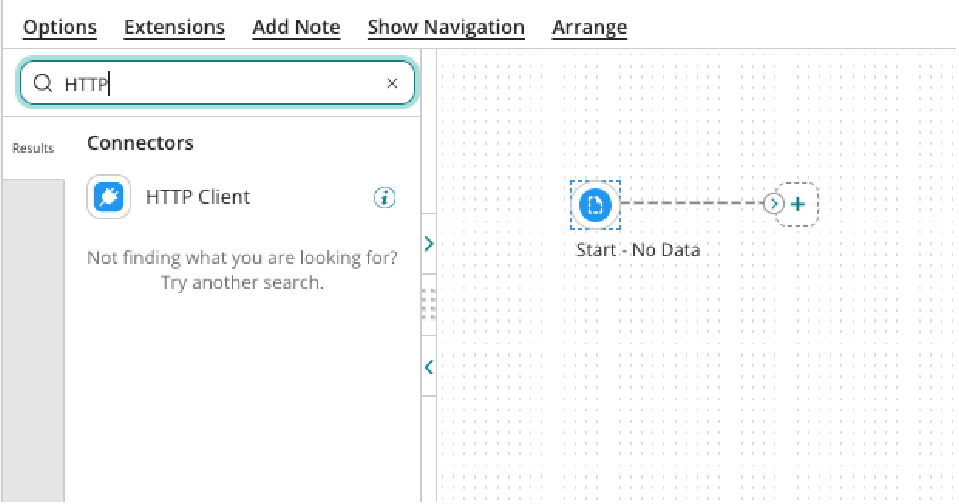
Implementing OAuth 2.0 with Client Credentials

In this blog, let us discuss about how to do a Manual OAuth call with client credentials grant type.

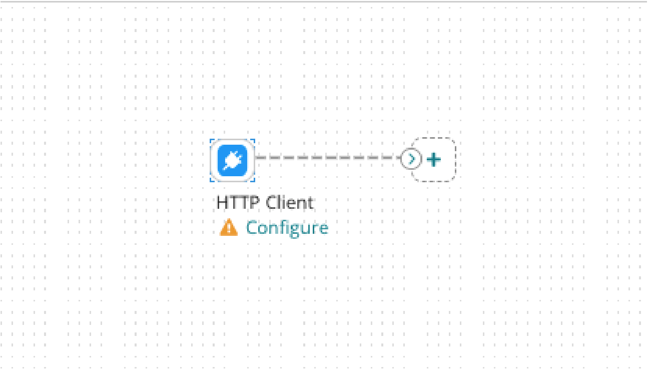
- To do manual OAuth call in boomi, we have a connector called "HTTP Client" through which we can use the service provided by third party API.
- Here, we want to access particular employee data from salesforce API where we give id as an input and will fetch complete data of an employee as the response.

Now, let us see the steps of "how to do the manual OAuth call".

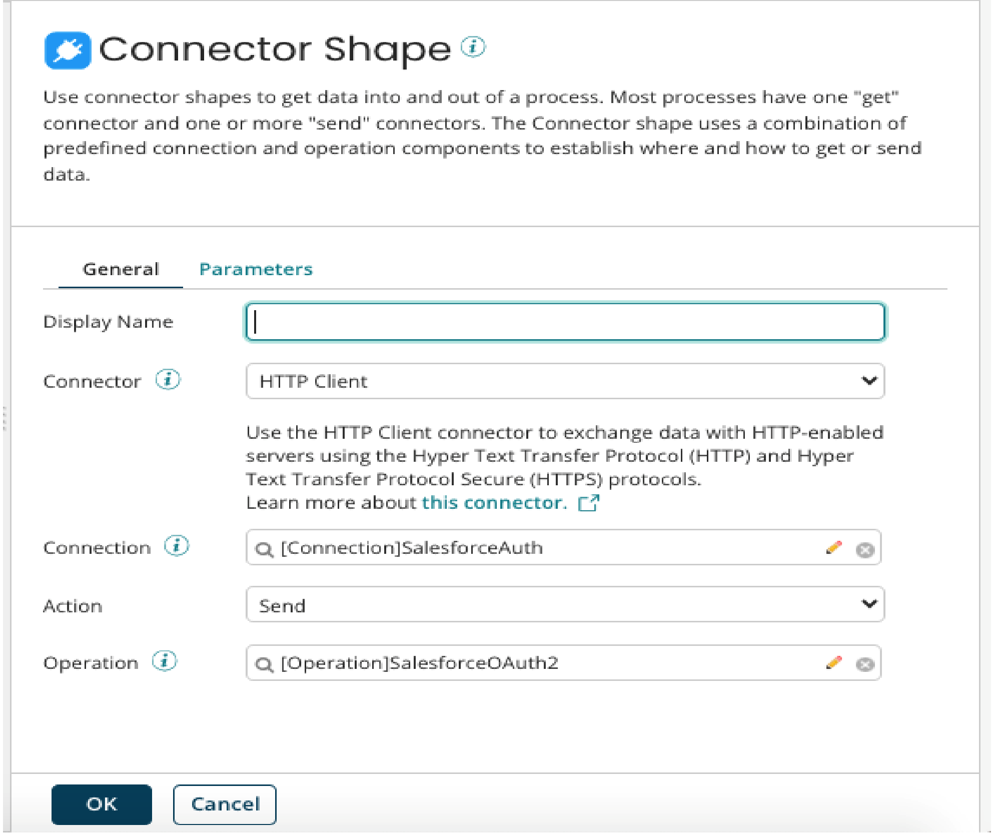
Step 1: Firstly, create a process with a start shape and type http in search map, you will get to see "Http Client" connector.



Step 2: Drag and drop "Http Client" connector onto the process canvas.



Step 3: Now, we have to configure 3 things in the connector i.e Action, Connector and Operation.



Connector Shape

Use connector shapes to get data into and out of a process. Most processes have one "get" connector and one or more "send" connectors. The Connector shape uses a combination of predefined connection and operation components to establish where and how to get or send data.

General | Parameters

Display Name:

Connector: HTTP Client

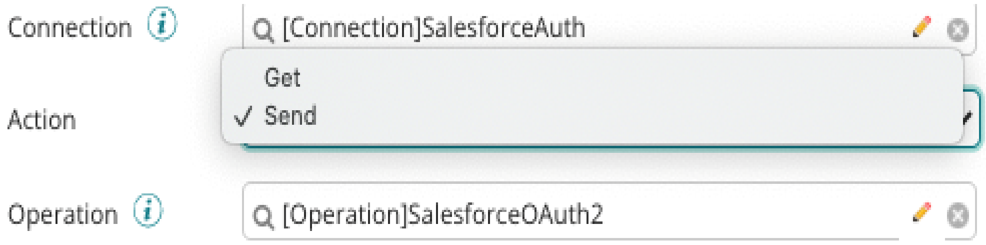
Use the HTTP Client connector to exchange data with HTTP-enabled servers using the Hyper Text Transfer Protocol (HTTP) and Hyper Text Transfer Protocol Secure (HTTPS) protocols.
Learn more about this connector.

Connection: [Connection]SalesforceAuth

Action: Send

Operation: [Operation]SalesforceOAuth2

Step 4: If we look at the Action, we find two operations i.e Get and Send. If we are sending data to an API, we will use send operation.



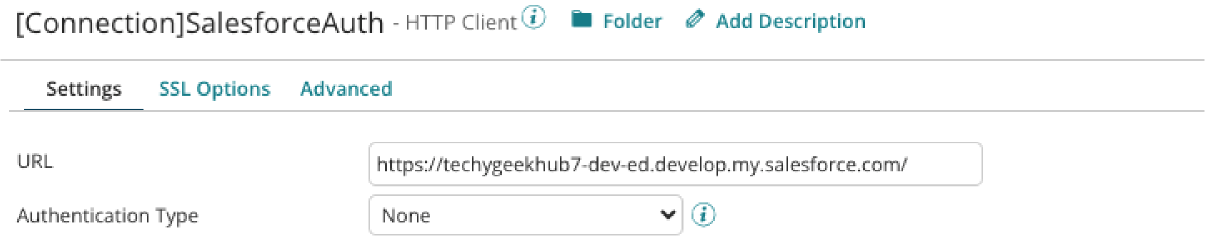
Connection: [Connection]SalesforceAuth

Action:

Operation: [Operation]SalesforceOAuth2

Step 5: To configure the connection, Click + and give a relevant name to the connection. You see 2 parts in the connection tab i.e URL and Authentication type.

- In place of URL, give base url of an api for which you want to consume the service.
- Here we will call the salesforce URL configured in the salesforce connected app in order to get the access token.
- We need to configure the connected app callback URL as base url, in this secanrio which is “https://techygeekhub7-dev-ed.develop.my.salesforce.com/”
- Based on the authentication type provided by the third party api, we have to choose the field. We are going to leave it as none which is default (as we don’t have any authentication)
- Click on save and close.



[Connection]SalesforceAuth - HTTP Client

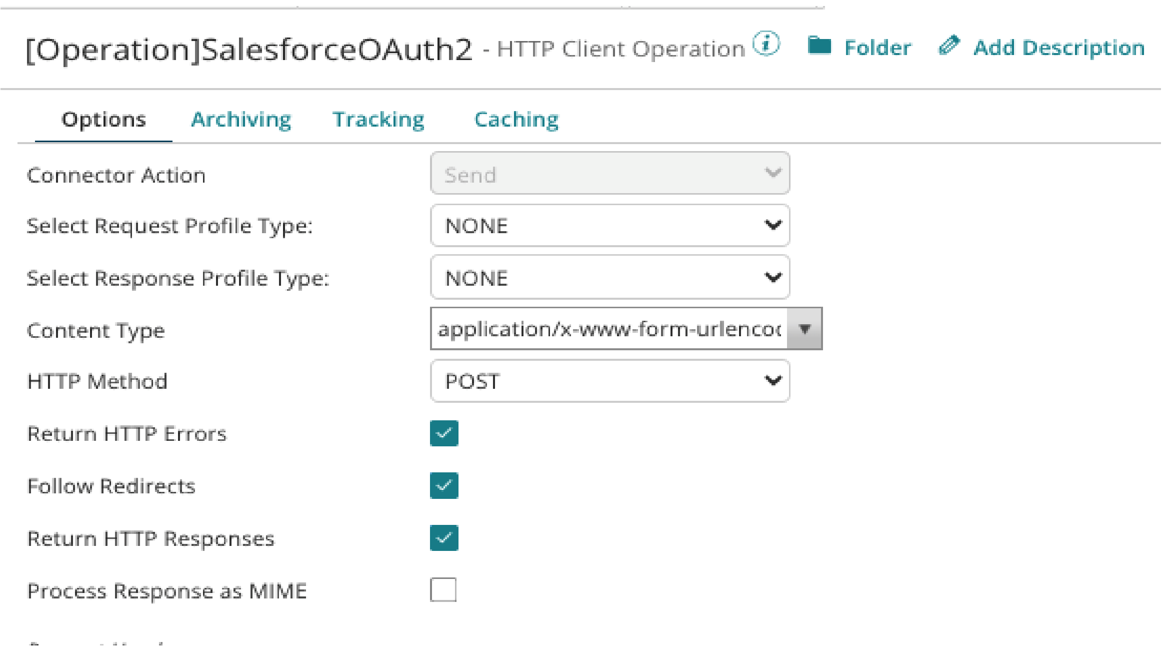
URL: https://techygeekhub7-dev-ed.develop.my.salesforce.com/

Authentication Type: None

Step 6: Now, click on + to configure the operation and give a relevant name.

In options tab, we find connector action as GET by default ,we have to configure SEND in the connector.

- Select request and response profiles based on the data that is being sent and received by the connector. we will put both request and response profiles as NONE as we don’t need any request or response profiles to access the API as we would only want records to be retrieved.
- Content type describes the format of how data is represented. In this example, it is application/x-www-form-urlencoded.
- HTTP Method defines the type of request. In this case, it would be SEND as we are sending some required data to get back the token from token server.



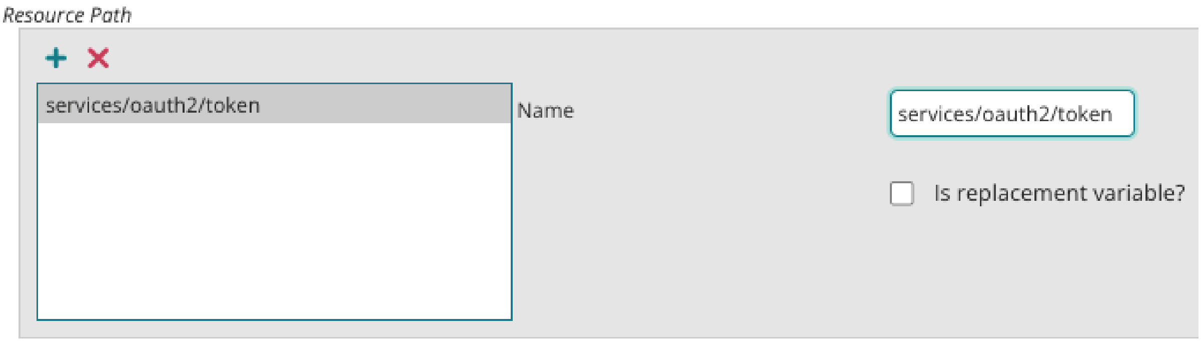
[Operation]SalesforceOAuth2 - HTTP Client Operation

Options | Archiving | Tracking | Caching

Connector Action	Send
Select Request Profile Type:	NONE
Select Response Profile Type:	NONE
Content Type	application/x-www-form-urlencoc
HTTP Method	POST
Return HTTP Errors	☑
Follow Redirects	☑
Return HTTP Responses	☑
Process Response as MIME	□

Step 7:

- If you scroll down you will find Request Headers where you can add the headers which are required to access the API. In this case, we don't have any headers.
- Next, place the resource path i.e (add the path which is after the base path).
- For suppose, the URL which we are going to access is "https://techygeekhub7-dev-ed.develop.my.salesforce.com/". So, "https://techygeekhub7-dev-ed.develop.my.salesforce.com/" becomes the base path, and "services/oauth2/token" will be added to the resource path.
- By calling this endpoint we will get back the access token.



Resource Path

services/oauth2/token

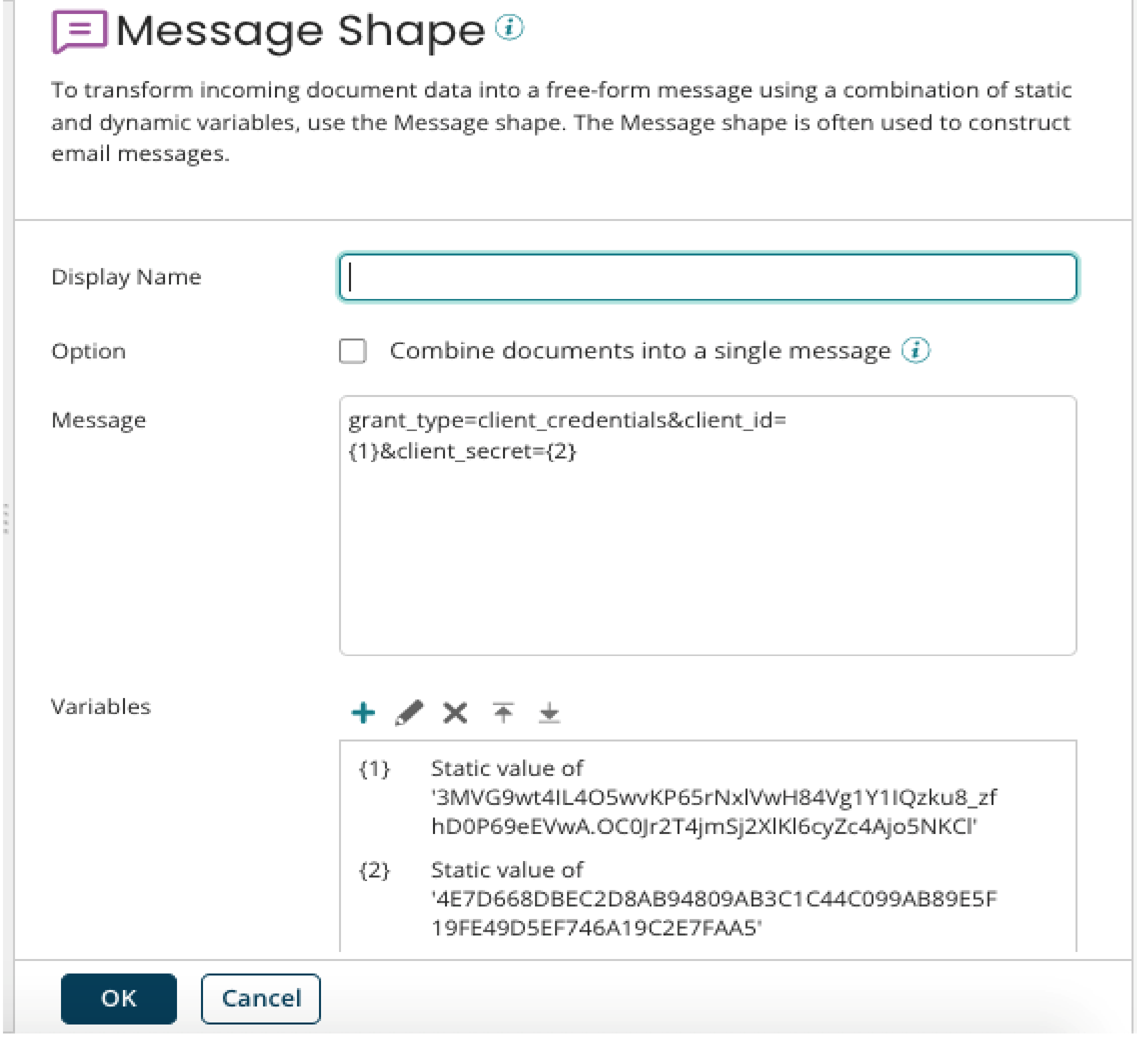
Name: services/oauth2/token

□ Is replacement variable?

Click save and close

Step 8: We need to drag and drop a message shape to pass the required information to the HTTP Client Connector.

- In the message shape we will pass grant_type=client_credentials&client_id={1}& client_secret={2}.
- The 1st place holder {1} will carry the value of client id and the 2nd place holder {2} will carry the value of the client secret which we can find from the salesforce connected app.



Message Shape

To transform incoming document data into a free-form message using a combination of static and dynamic variables, use the Message shape. The Message shape is often used to construct email messages.

Display Name

Option □ Combine documents into a single message

Message

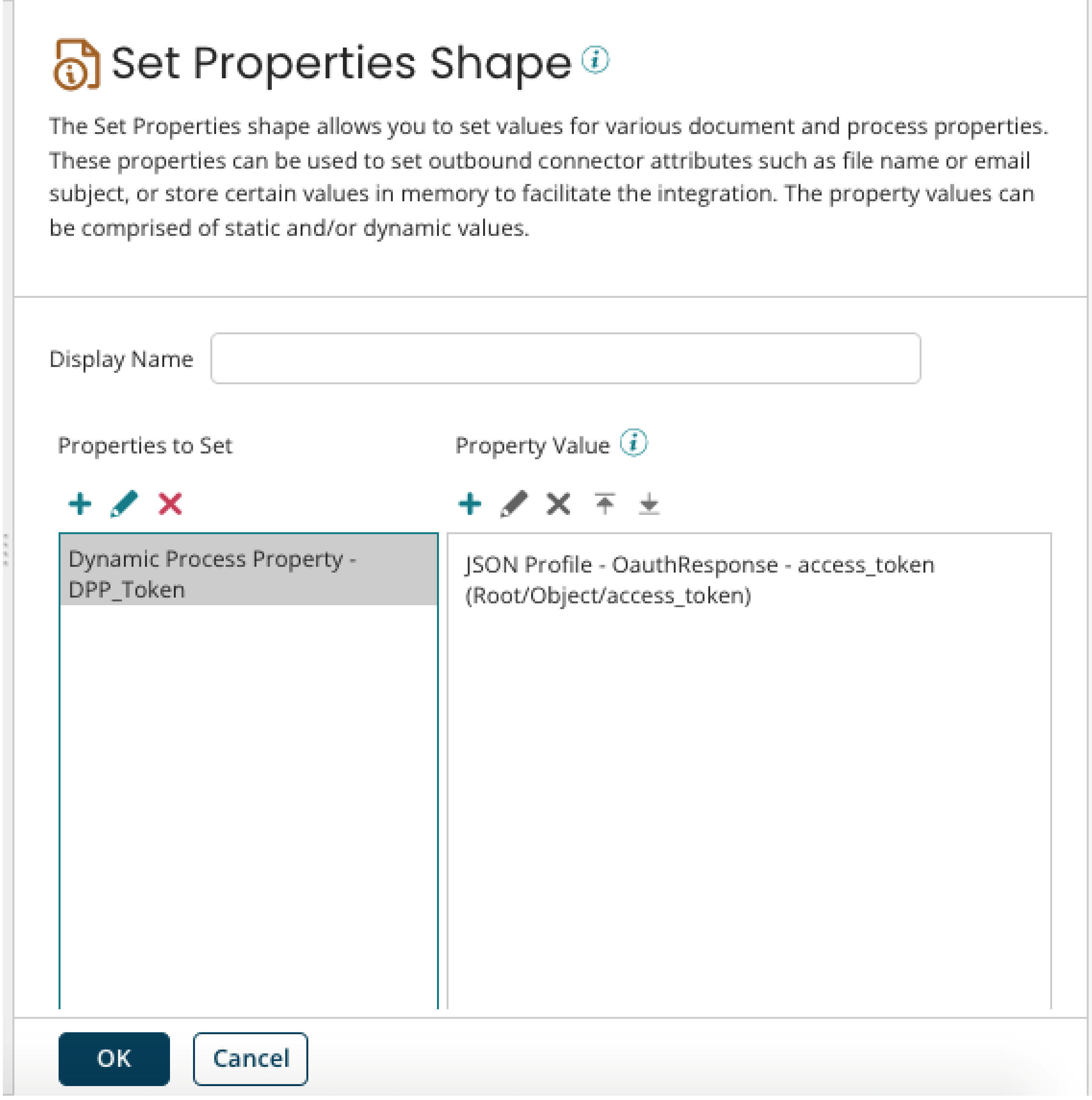
```
grant_type=client_credentials&client_id=
{1}&client_secret={2}
```

Variables

{1} Static value of '3MVG9wt4IL4O5wvKP65rNxlVwH84Vg1Y1IQzku8_zfhD0P69eEVwA.OC0Jr2T4jmSj2XlKl6cyZc4Ajo5NKCl'

{2} Static value of '4E7D668DBEC2D8AB94809AB3C1C44C099AB89E5F19FE49D5EF746A19C2E7FAA5'

Step 9: Now we need to drag and drop a set properties shape. Then we need to setup a Dynamic Process Property to capture the Access Token.



Set Properties Shape

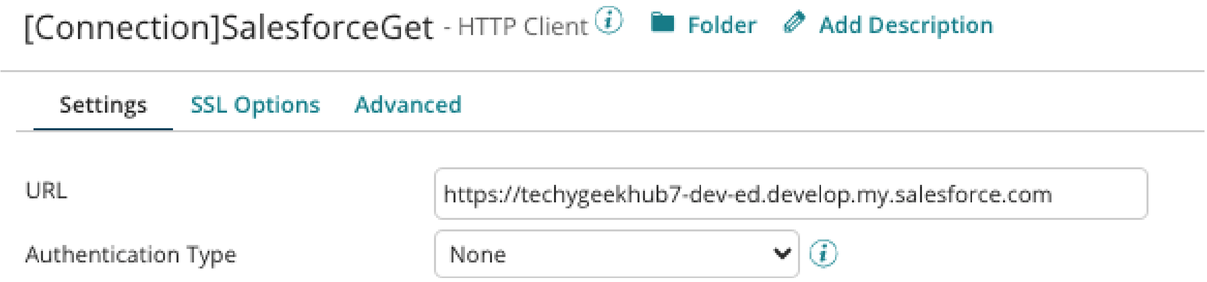
The Set Properties shape allows you to set values for various document and process properties. These properties can be used to set outbound connector attributes such as file name or email subject, or store certain values in memory to facilitate the integration. The property values can be comprised of static and/or dynamic values.

Display Name

Properties to Set	Property Value
Dynamic Process Property - DPP_Token	JSON Profile - OauthResponse - access_token (Root/Object/access_token)

Step 10: Now we need to drag and drop a branch shape and we connect the first branch to the above built logic and in the second branch we will built the logic to use the retrieved token and get the resource.

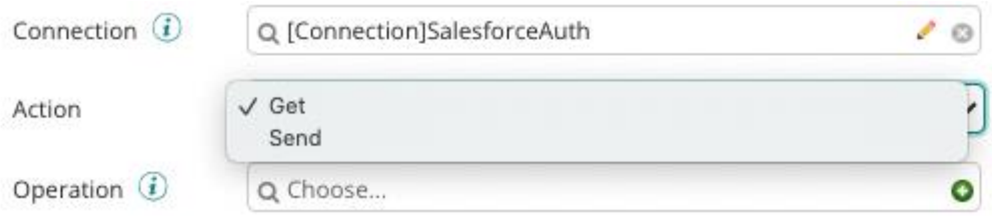
Step 11: Drag and drop a HTTP Connector and configure it with the Base URL as "https://techygeekhub7-dev-ed.develop.my.salesforce.com/".


[Connection]SalesforceGet - HTTP Client

URL: https://techygeekhub7-dev-ed.develop.my.salesforce.com

Authentication Type: None

Step 12: Now we need to configure the action as Get and we need to configure the operation in HTTP connector.

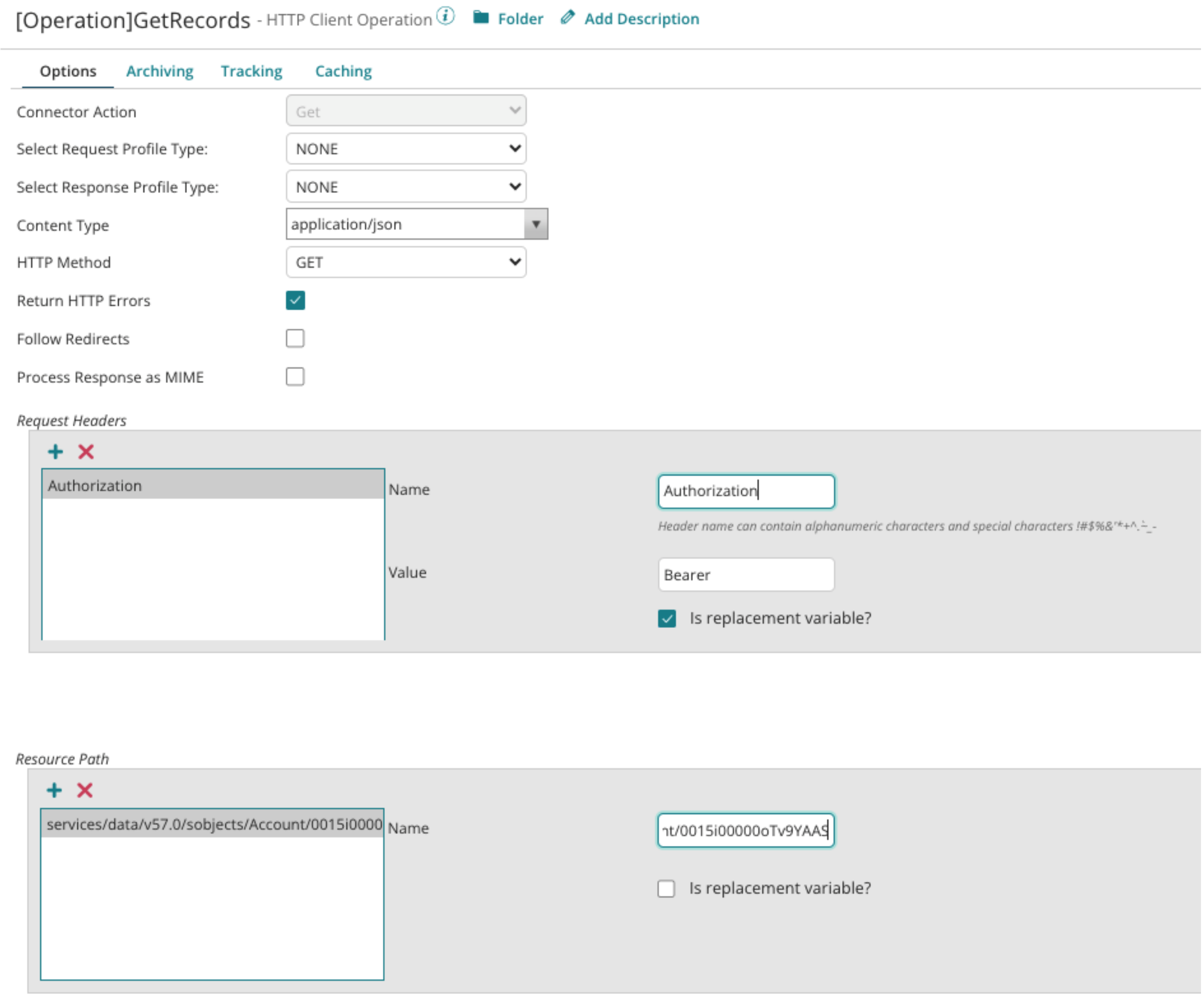

Connection: [Connection]SalesforceAuth

Action: ✓ Get

Operation: Choose...

Step 13: We need to configure the Request and Response profile. We will select as None.

- Then we need to configure the request header as we have to pass the access token in the header.
- To do so, we need to click on the plus icon and we need to add one header named “Authorization”. The value section we choose as a replacement variable as we will pass it from parameters.
- Then we will configure the resource path as “services/data/v57.0/sobjects/Account/0015i00000oTv9YAAS” to get back the resource.



[Operation]GetRecords - HTTP Client Operation

Connector Action: Get

Select Request Profile Type: NONE

Select Response Profile Type: NONE

Content Type: application/json

HTTP Method: GET

Return HTTP Errors ☑

Follow Redirects ☐

Process Response as MIME ☐

Request Headers

Authorization

Name: Authorization

Header name can contain alphanumeric characters and special characters !#$%&'+^.-_-*

Value: Bearer

☑ Is replacement variable?

Resource Path

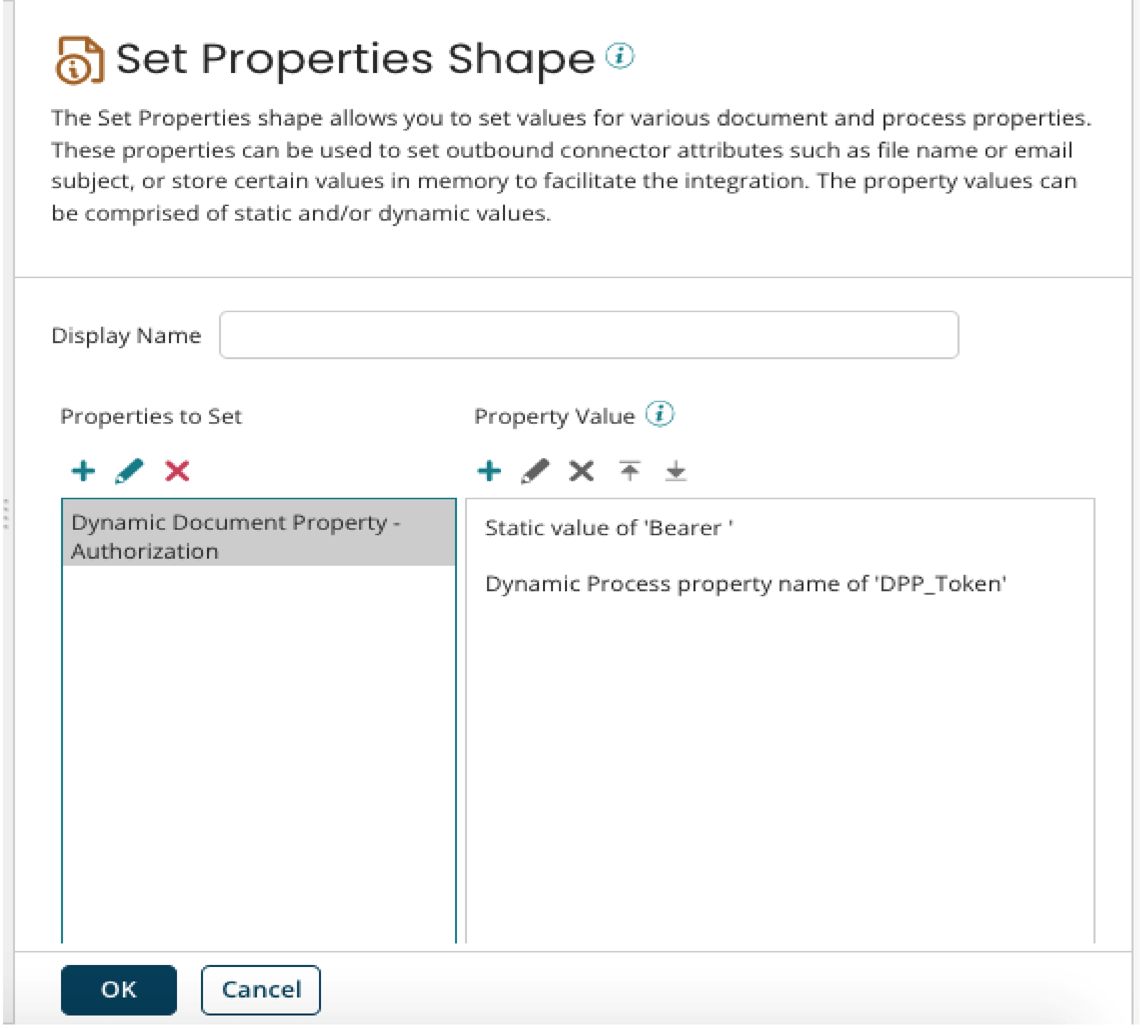
services/data/v57.0/sobjects/Account/0015i0000

Name: nt/0015i00000oTv9YAAS

☐ Is replacement variable?

Step 14: To pass the token value we need to drag and drop a set properties before the HTTP connector.

- On the set properties we need to configure a Dynamic Document Property same as the replacement variable name of the request header name of the HTTP connector operation.
- We need to pass the Dynamic process property value configured before which we used to capture the token.



Set Properties Shape

The Set Properties shape allows you to set values for various document and process properties. These properties can be used to set outbound connector attributes such as file name or email subject, or store certain values in memory to facilitate the integration. The property values can be comprised of static and/or dynamic values.

Display Name

Properties to Set

Dynamic Document Property - Authorization

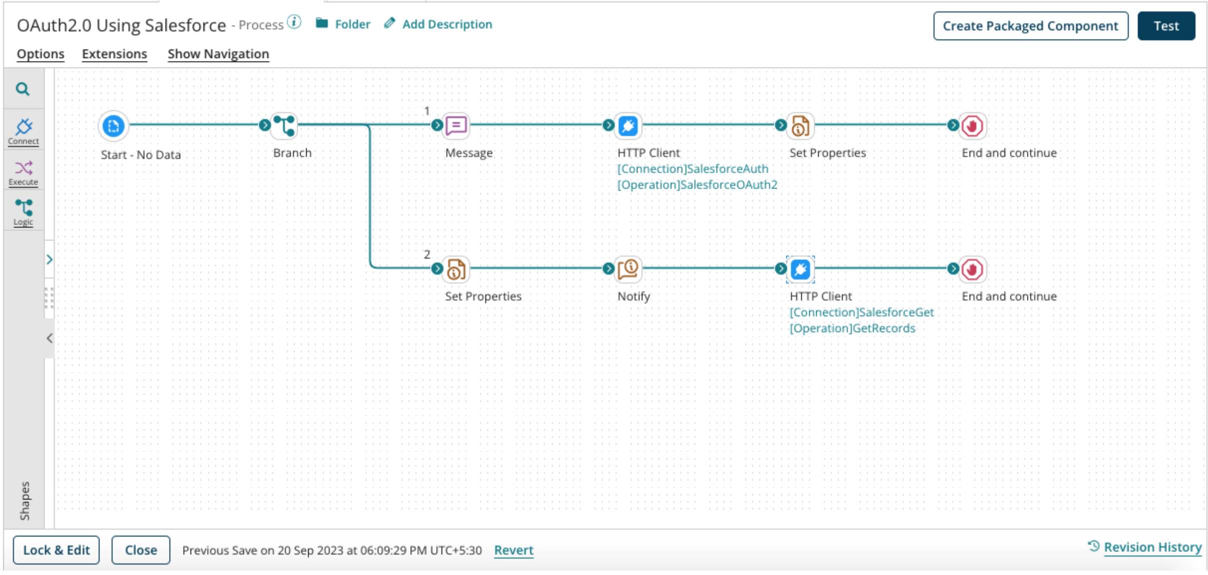
Property Value

Static value of 'Bearer '

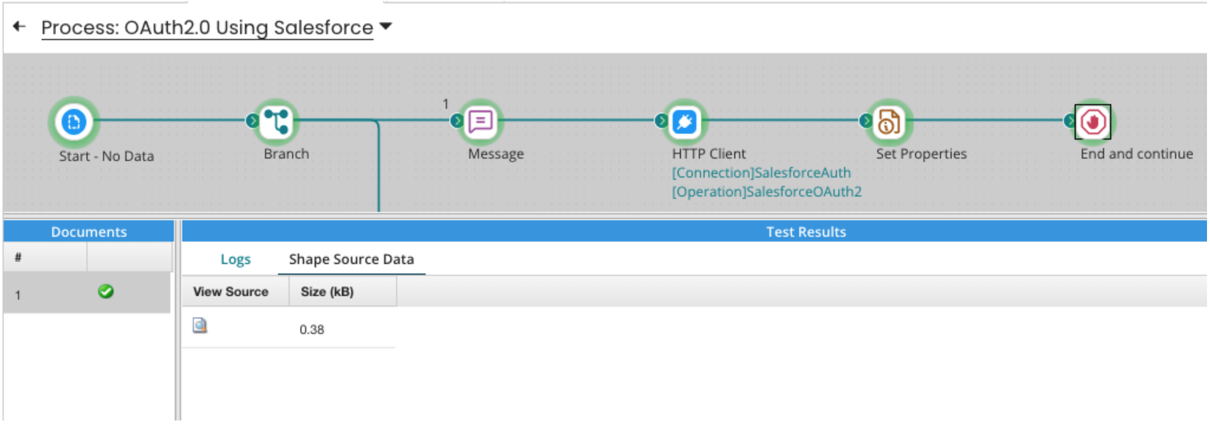
Dynamic Process property name of 'DPP_Token'

Step 15: We need to drag and drop a STOP shape to stop the flow of execution.

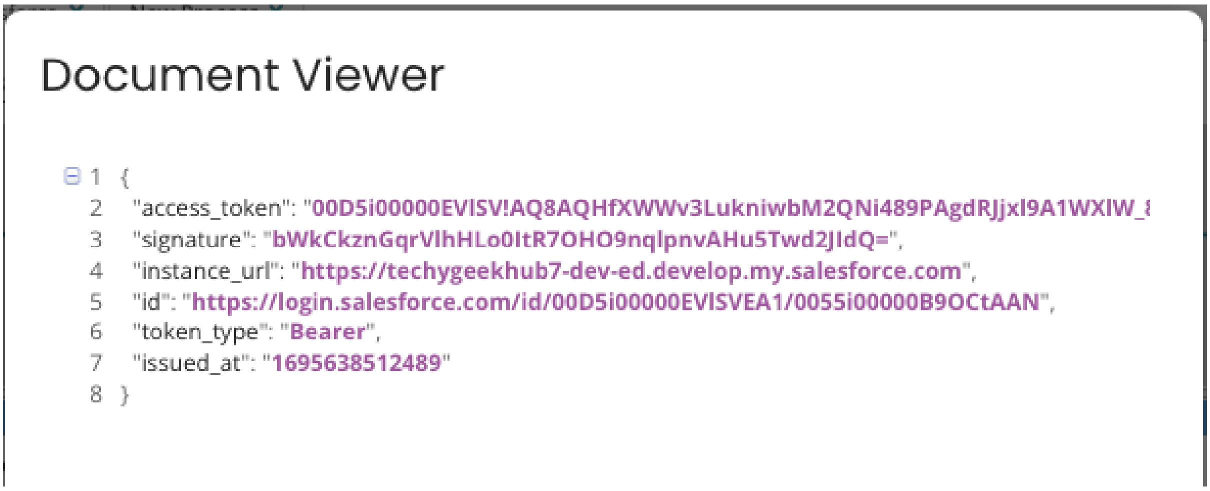
After configuring everything our process will look something like shown in the below.



Step 16: Now we need to test the process and we can get the resource from the salesforce as shown below.

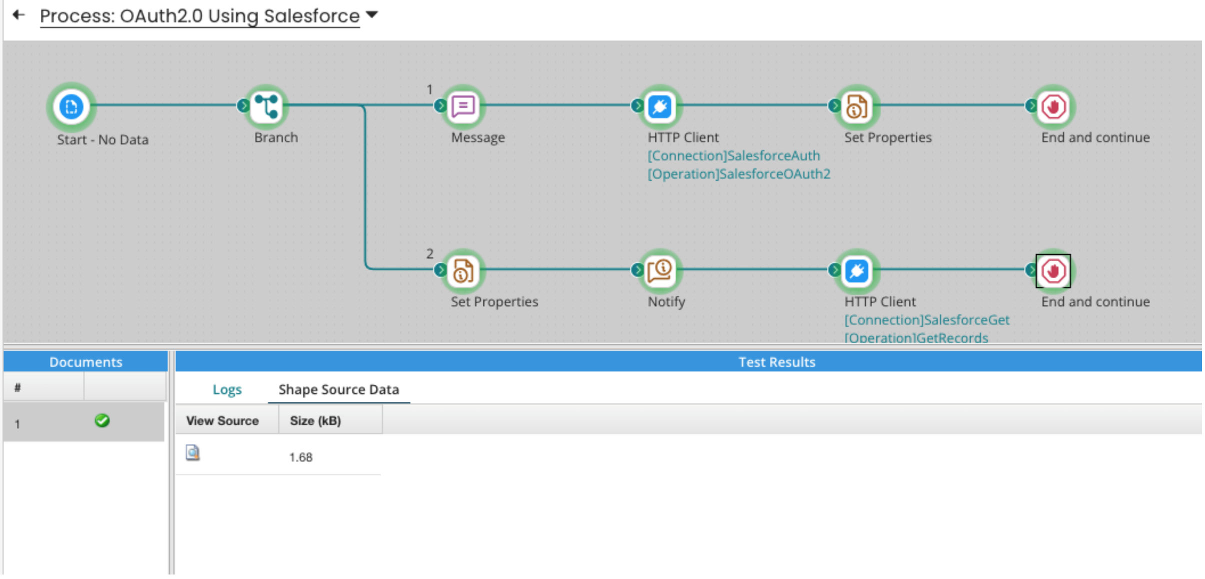


#		View Source	Size (kB)
1	✓		0.38

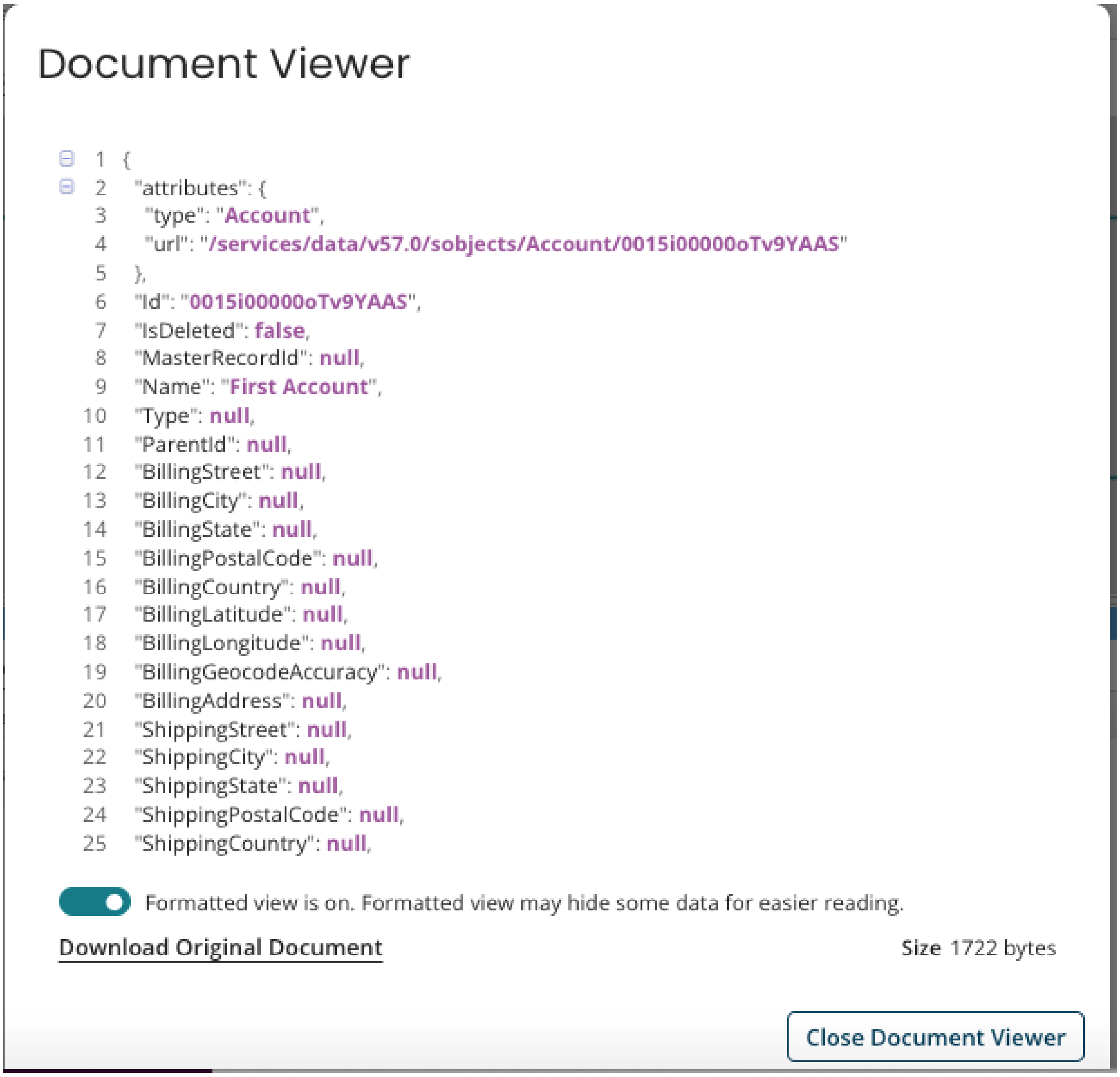


Document Viewer

```
{
  "access_token": "00D5i00000EVlSV!AQ8AQHfXWWv3LukniwbM2QNi489PAgdRJjxl9A1WXlW_
  "signature": "bWkCkznGqrVlhHLo0ItR7OHO9nqlpnvAHu5Twd2JIdQ=",
  "instance_url": "https://techygeekhub7-dev-ed.develop.my.salesforce.com",
  "id": "https://login.salesforce.com/id/00D5i00000EVlSVEA1/0055i00000B9OCtAAN",
  "token_type": "Bearer",
  "issued_at": "1695638512489"
}
```



#		View Source	Size (kB)
1	✓		1.68



Document Viewer

```
{
  "attributes": {
    "type": "Account",
    "url": "/services/data/v57.0/sobjects/Account/0015i00000oTv9YAAS"
  },
  "Id": "0015i00000oTv9YAAS",
  "IsDeleted": false,
  "MasterRecordId": null,
  "Name": "First Account",
  "Type": null,
  "ParentId": null,
  "BillingStreet": null,
  "BillingCity": null,
  "BillingState": null,
  "BillingPostalCode": null,
  "BillingCountry": null,
  "BillingLatitude": null,
  "BillingLongitude": null,
  "BillingGeocodeAccuracy": null,
  "BillingAddress": null,
  "ShippingStreet": null,
  "ShippingCity": null,
  "ShippingState": null,
  "ShippingPostalCode": null,
  "ShippingCountry": null,
```

Formatted view is on. Formatted view may hide some data for easier reading.

Download Original Document

Size 1722 bytes

TGH

Making Integrations Simpler

boomi Partner

TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.

Email address

connect@techygeekhub.com

Phone number

+ 011-40071137
+ 91-8810610395

Our offices

Noida Office
iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office
Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081