



TGH

Making Integrations Simpler

boomi
Partner



Integration with Slack in Boomi



Integration with Slack in Boomi

In this blog, we will see how to leverage Slack connector using Web Listener Service in Boomi.

What is Slack?

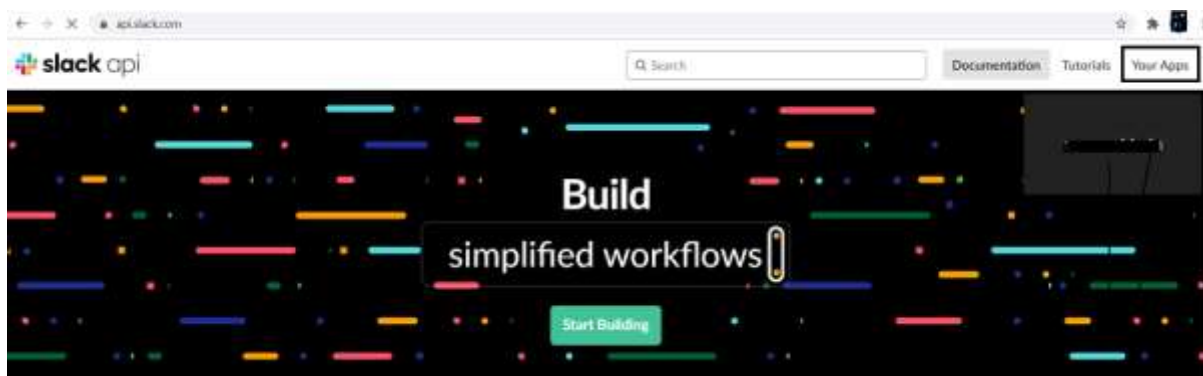
Slack is a messaging application where team members can communicate and work together more effectively. It allows us to chat in groups, and upload and share files with team members. Slack teams allow communities, groups or teams to join a workspace via the specific URL or invitation sent by a team admin or owner.

In this Use Case, we will configure a listener process which accepts JSON payload and stores it in dynamic document property calls it in a map and sends that data into Slack.

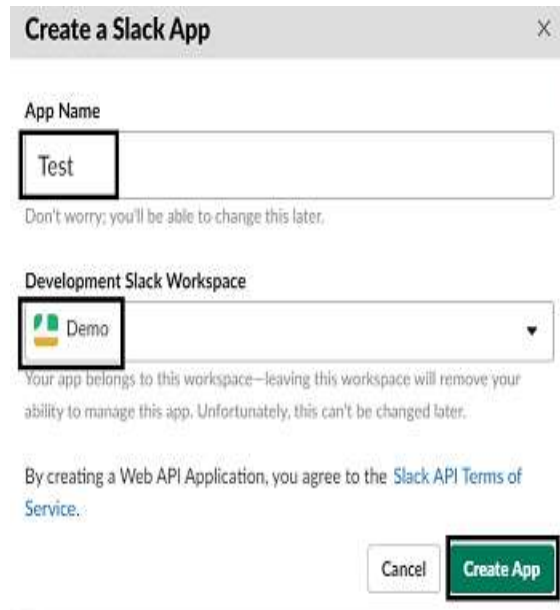
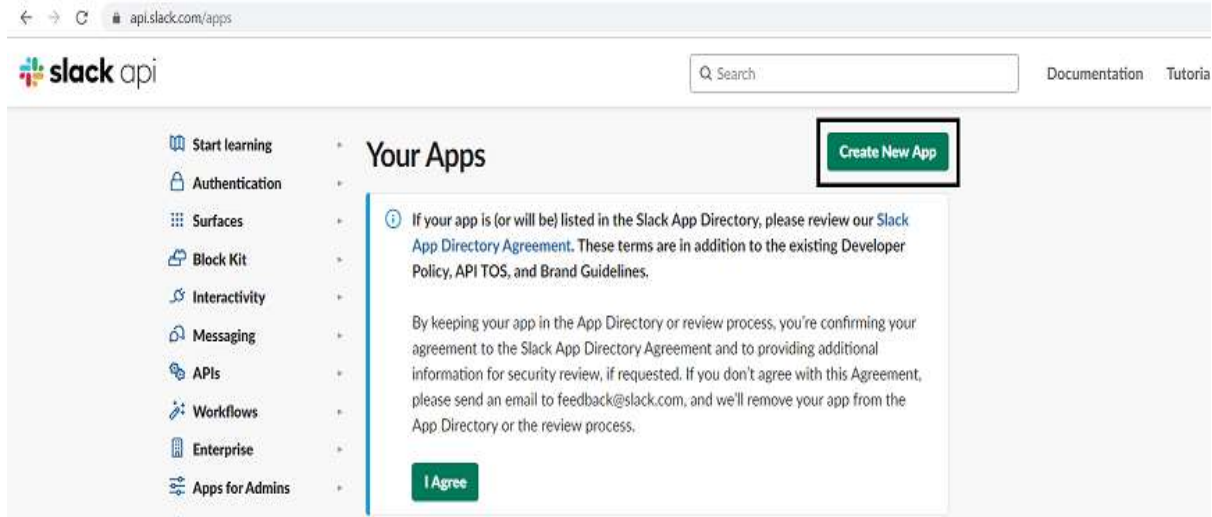
NOTE: Assuming that we have a Slack account and Channel created, we will create the application in Slack from where we get Client ID and Client Secret which is required for establishing a Slack connection in Boomi.

Let us first create the application in Slack.

Step 1: Go to <https://api.slack.com/> and click on Your Apps

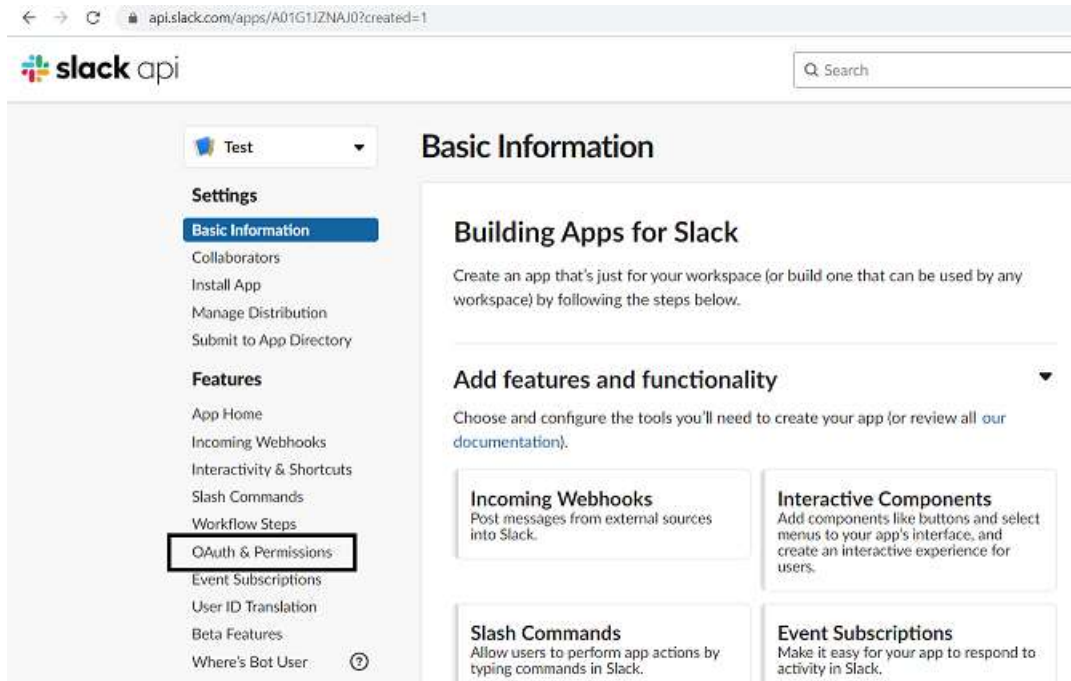


Step 2: Click on Create New App give a name to the App and select the workspace from the drop-down which we have given while creating the account. Click on Create App

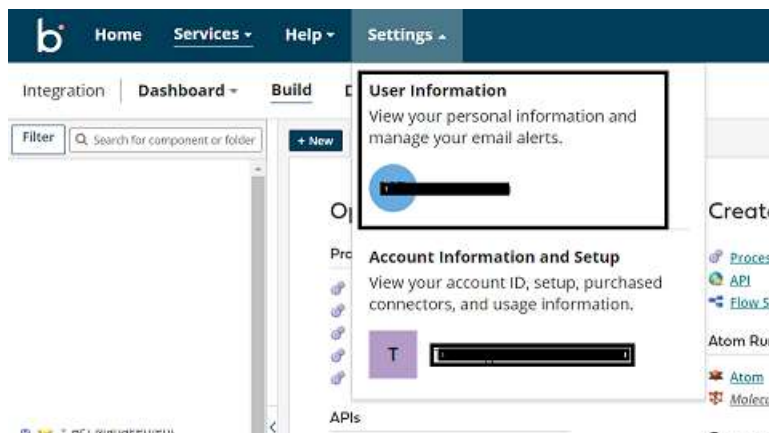


The screenshot shows the 'Create a Slack App' dialog box. The 'App Name' field contains 'Test' and the 'Development Slack Workspace' dropdown is set to 'Demo'. 'Create App' and 'Cancel' buttons are at the bottom.

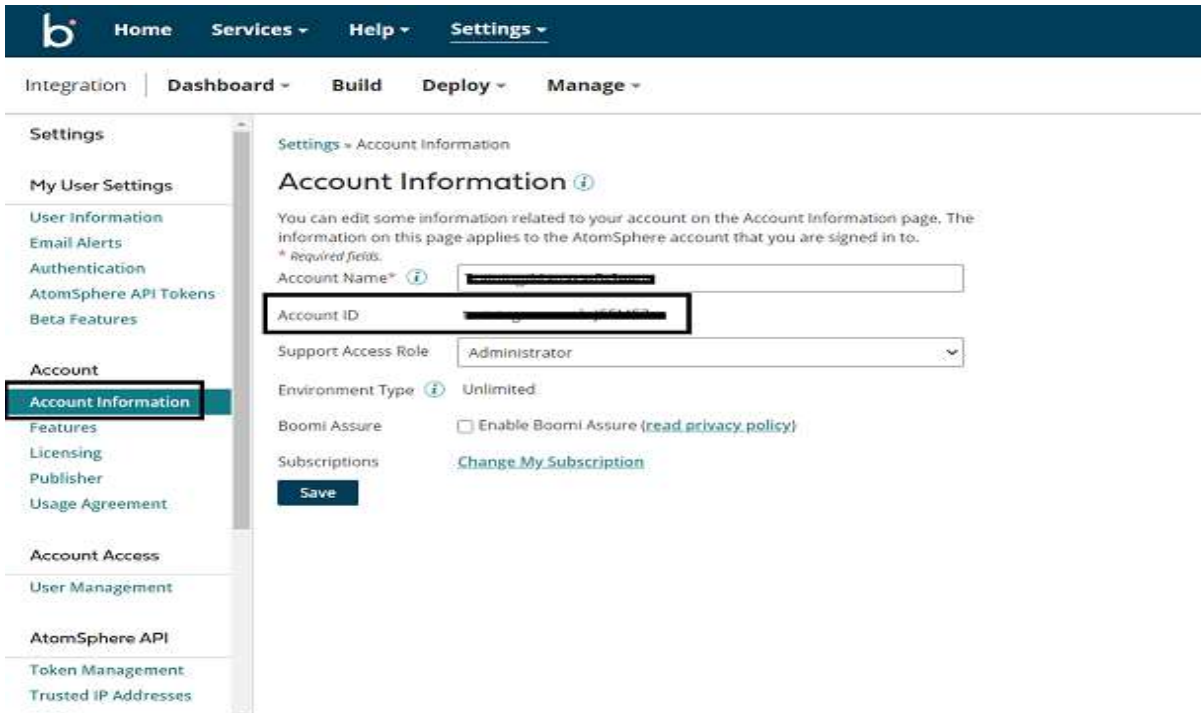
Step 3: Select OAuth and Permissions to set redirect Url and get a Client ID and Client Secret.



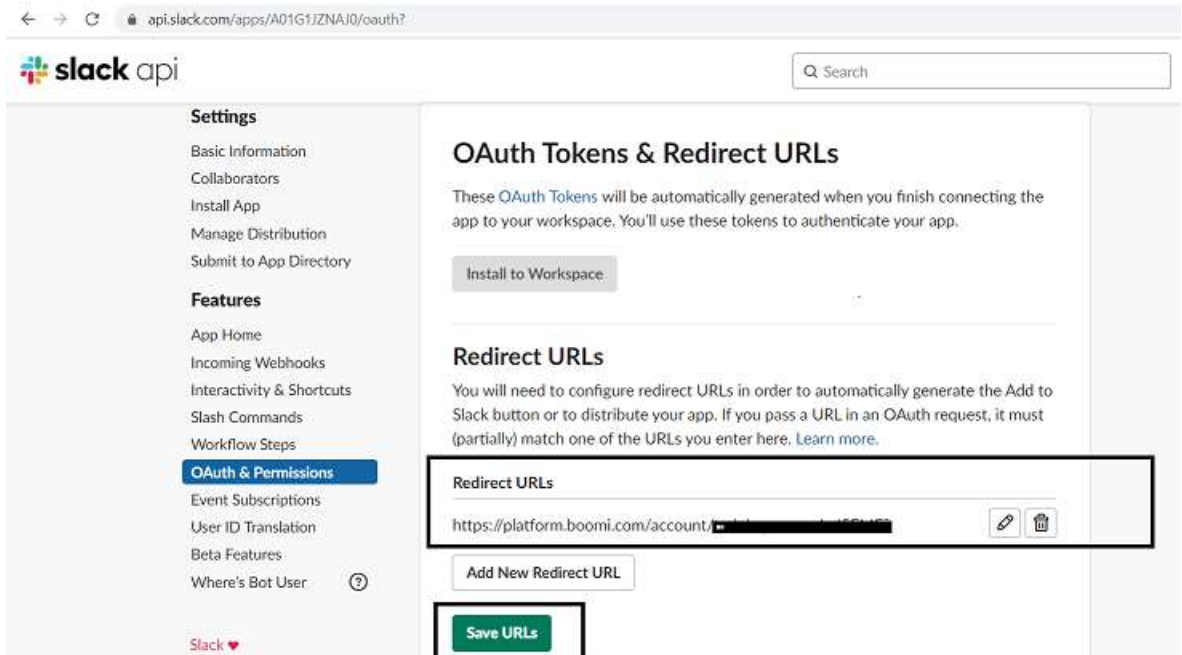
- For OAuth 2.0 authentication purposes, we set OAuth 2.0 redirect URI to **[https://platform.boomi.com/account/<your_Boomi Integration_account_ID>/oauth2/callback](https://platform.boomi.com/account/<your_Boomi_Integration_account_ID>/oauth2/callback)**.
- **Give your Boomi Account ID in place of <your_Boomi Integration_account_ID> in the above URL.**
- To get a Boomi Account ID, log into the Boomi Platform. Go to settings and click on User Information.



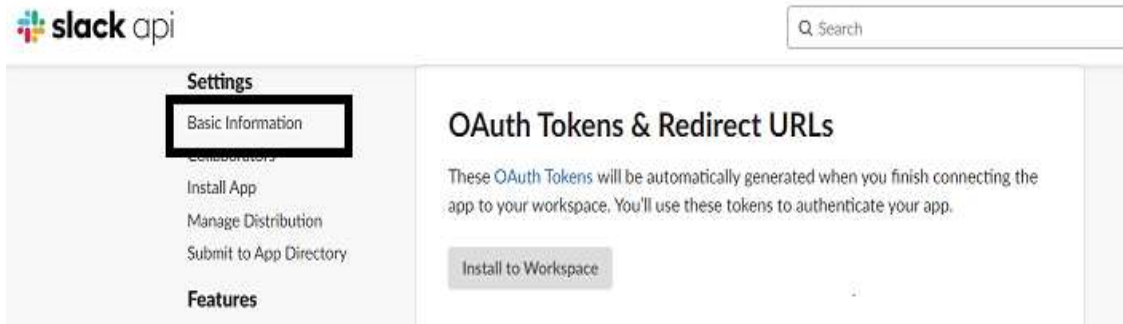
- Select Account Information on the left and we will find Account ID



Step 4: Now, set the Redirect Uri to <https://platform.boomi.com/account/<account ID>/oauth2/callback> in the Oauth page by clicking on the add redirect URL. Click on Save URLs.



Step 5: Go to Basic Information to get the client ID and Secret Key.

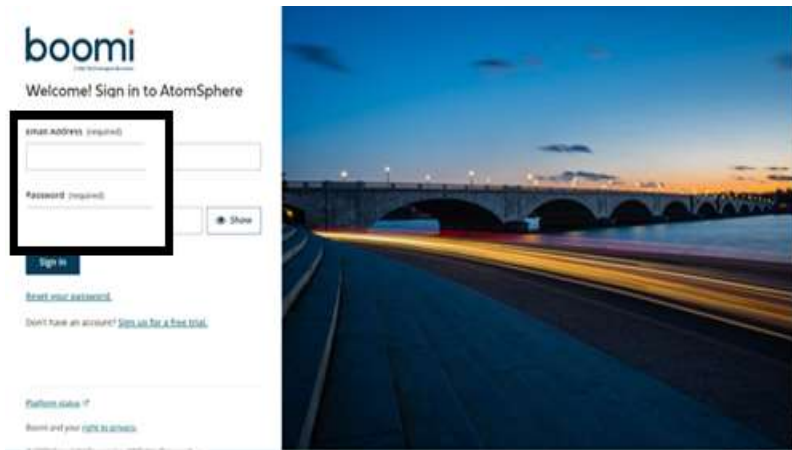


Step 6: Once we scroll down, we see the client ID and Client Secret which will be used to create a Slack connection in Boomi.

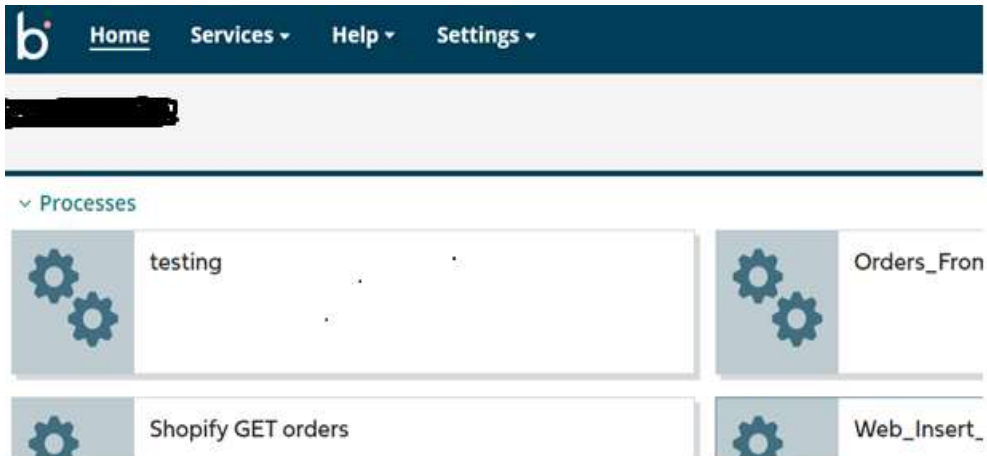


Now that we have the client ID and client secret, we will create the process to integrate Slack Connector in Boomi.

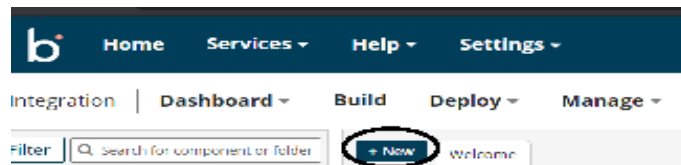
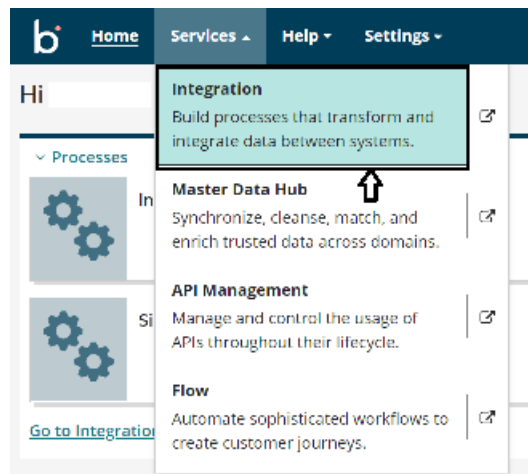
Step 1: Log on to the Boomi platform (<https://platform.boomi.com/>) with the required credentials i.e. Email Address and Password.



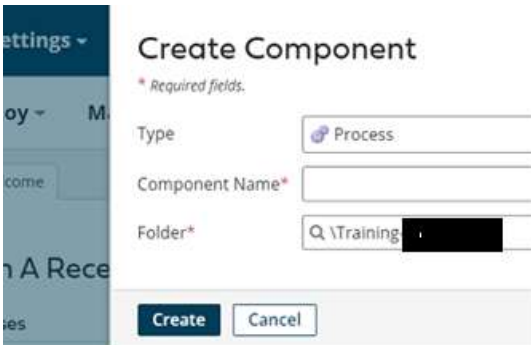
Step 2: Once logged into the Boomi platform, we will be able to view the Home page.



Step 3: Now, click on Services followed by Integration. We will see the Build page. Click on New.



Step 4: Once, click on New, we will be able to see three fields i.e. Type, Component Name and Folder.

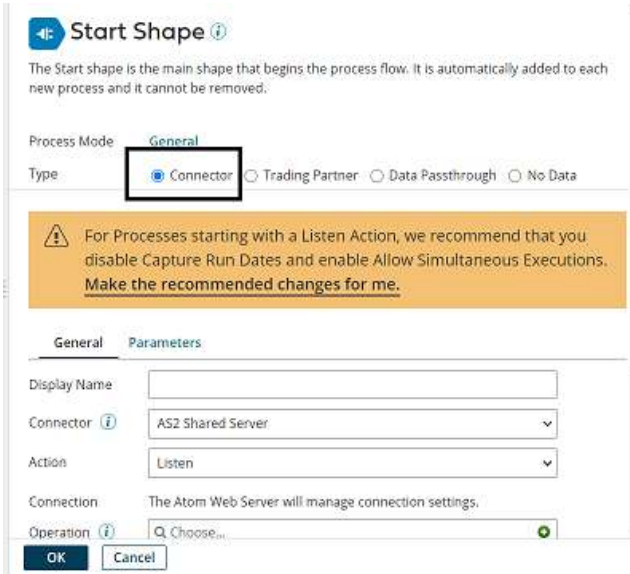


- Select Type as process as we are building a process. Component Name and Folder can be given based on your choice (i.e. which name to be given and where do we want to create the process). Click on Create.

Step 5: We see that the process gets created with a start shape which is configured with AS2 Shared Server by default.



Step 6: Select the start shape and choose the type as Connector as we are configuring the Listener Process.



Step 7: Change Connector to Web Services Server Connector and action will be Listen by default. Click + on operation and give it a name.



Slack Listener Operation - Web Services Server Operation ⓘ

Options Archiving Tracking

Connector Action Listen ▾

Simple URL Path ⓘ /ws/simple/get

SOAP URL Path ⓘ /ws/soap

Operation Type ⓘ GET ▾

Object ⓘ

Expected Input Type ⓘ None ▾

Response Output Type ⓘ None ▾

Attachment Cache ⓘ Q Choose... +

Step 8: Change Operation Type to CREATE and add the name of the object to slackdemo.

Slack Listener Operation - Web Services Server Operation ⓘ

Options Archiving Tracking

Connector Action Listen ▾

Simple URL Path ⓘ /ws/simple/createSlackdemo

SOAP URL Path ⓘ /ws/soap

Operation Type ⓘ CREATE ▾

Object ⓘ slackdemo

Expected Input Type ⓘ None ▾

Response Output Type ⓘ None ▾

Attachment Cache ⓘ Q Choose... +

Step 9: Here, the Expected Input Type will be a Single Json Object and we have saved the sample file in the local directory and will upload the file by clicking + on the request profile. Name it and click on import.

Account JSON Profile JSON Profile ⓘ Folder Add Description

Import

Getting Started

JSON Profiles

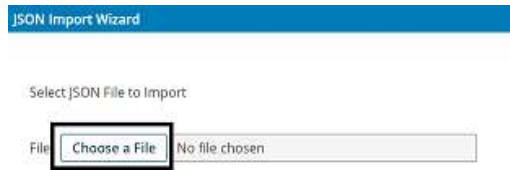
JSON profiles represent the structure of the various JSON documents sent through Boomi processes, including all the elements, attributes, and other configuration information. The JSON profile allows you to configure repeating data sets.

You can import from a file or one of the JSON template standards that Boomi provides. Would you like to import your profile from one of these sources now?

Import a Profile

↑

Step 10: Choose the file from the local directory.



Step 11: Here, it will be Account.json. Click Next and Finish.



Step 12: We see that the profile has been loaded all the fields got imported.



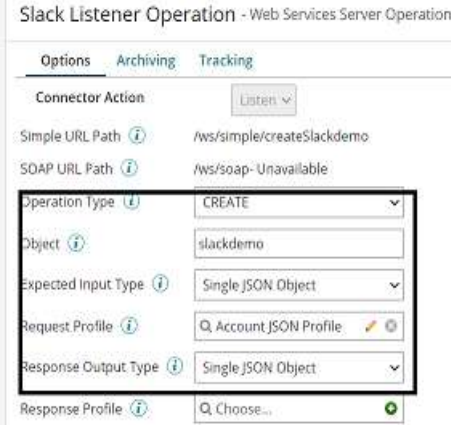
- The sample JSON file looks like this,

```
{
  "Id" : "1000200300400",
  "Street" : "Kings Park, 17th Avenue",
  "State" : "UK",
  "AccountNumber" : "CD355119-A",
  "Website" : "http://www.ioc.com",
  "Industry" : "Energy",
  "NumberOfEmployees" : "2400",
  "Ownership" : "public"
}
```



- Click on save and close.

Step 13: Select the expected input type as a single JSON Object and the Response profile will be given in a message body of message shape.



Slack Listener Operation - Web Services Server Operation

Options Archiving Tracking

Connector Action Listen

Simple URL Path /aws/simple/createSlackdemo

SOAP URL Path /aws/soap- Unavailable

Operation Type CREATE

Object slackdemo

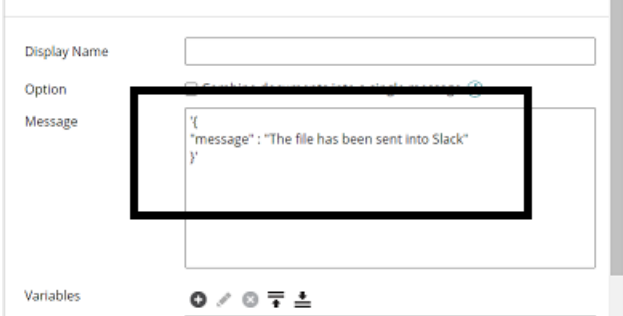
Expected Input Type Single JSON Object

Request Profile Q, Account JSON Profile

Response Output Type Single JSON Object

Response Profile Q, Choose...

Step 14: Drag and drop the message shape to add the response profile. Add the JSON message as shown in the Message.



Display Name

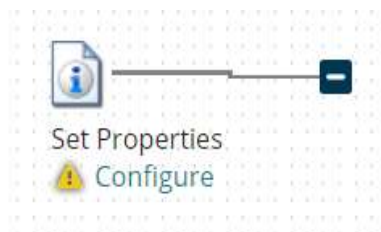
Option

Message

```
{
  "message": "The file has been sent into Slack"
}
```

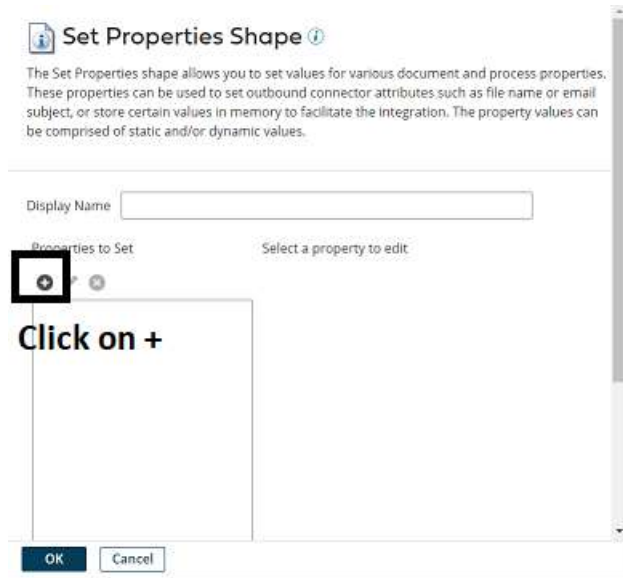
Variables

Step 15: Drag and drop set properties shape on process canvas to set Dynamic Document property which holds the current data.

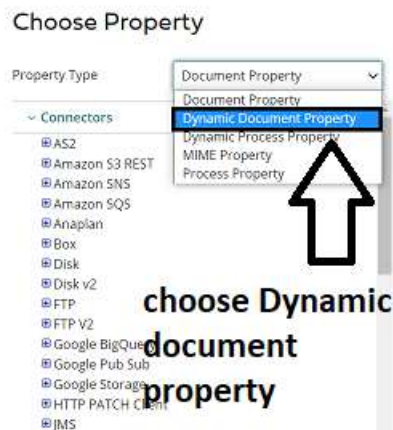


Step 16: We will now create dynamic document properties in set properties shape.

- Let us create by selecting the set properties shape and clicking on the + symbol as shown.



- Choose Dynamic Document Property from the drop-down.



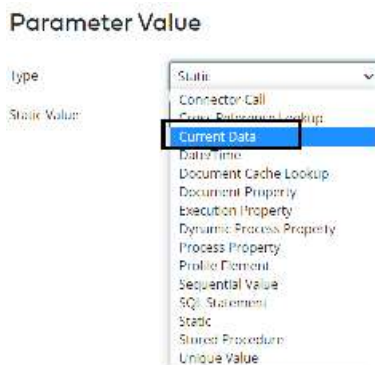
- Give the property name as “Text Data” and click ok.



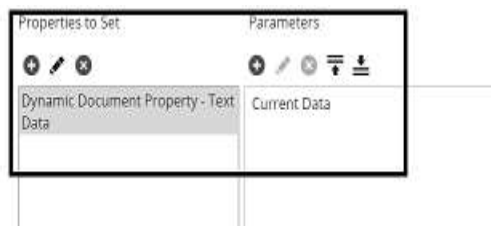
- Now, we have to set parameters for Dynamic Document Property. Select the Dynamic Document property and choose + on parameters.



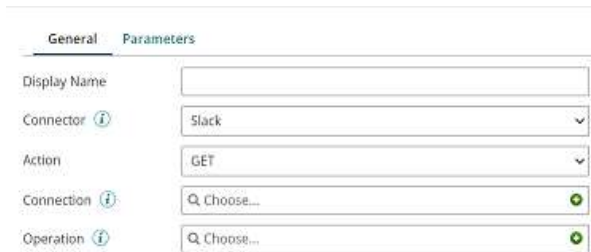
- We will select Type as current data. Click ok.



Step 17: After setting the property and parameter in the set properties shape, it looks like



Step 18: Drag and drop the slack connector onto the process canvas and configure it.



The screenshot shows a configuration form with two tabs: 'General' and 'Parameters'. The 'Parameters' tab is active. It contains the following fields:

- Display Name:
- Connector:
- Action:
- Connection:
- Operation:

Step 19: Here we will set the action as send as we are creating the message into Slack. Choose + on connection to create a new one. Select OAuth2.0.



The screenshot shows the 'Slack Connection' configuration form. The 'Connection' dropdown is set to 'OAuth 2.0'. The 'Authentication Type' dropdown is also set to 'OAuth 2.0'. There are buttons for 'Folder' and 'Add Description'.

Step 20: Give client ID and Client Secret which we have received after creating the app in Slack.



The screenshot shows the 'Slack Connection' configuration form. The 'Client ID' field is filled with a blacked-out value. The 'Client Secret' field is filled with '<Encrypted>'. There is a 'Generate...' button next to the 'Access Token' field.

Step 21: Click on Generate beside the access token



The screenshot shows the 'Slack Connection' configuration form. The 'Access Token' field is highlighted with a black box, and the 'Generate...' button next to it is also highlighted.

Step 22: Give your Slack workspace and follow the steps which are asked.



Sign in to your workspace

Enter your workspace's Slack URL

Continue

Don't know your workspace URL? [Find your workspaces](#)

Looking to create a workspace instead? [Create a new workspace](#)

- Give email address and password.



Sign in to Demo

[workspace].slack.com

 Continue with Google

OR

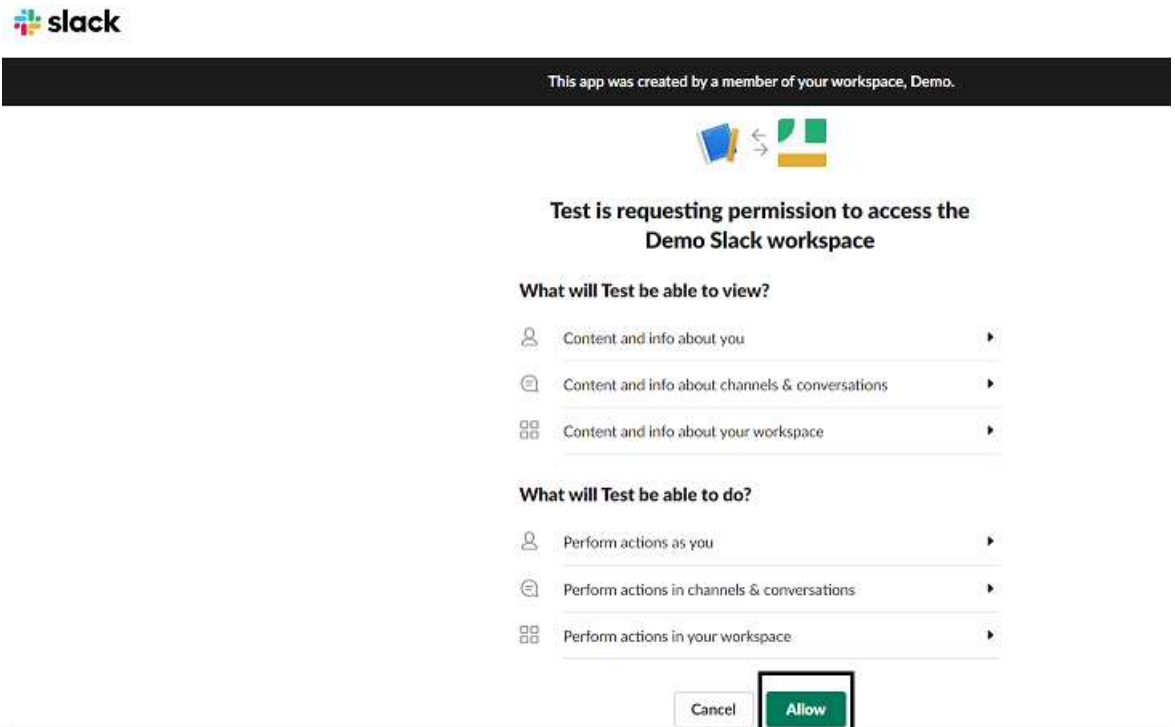
Email address

Password

Sign in

Remember me on this device

- Click on Allow which means that the third-party application can access your app.



- We see that the authentication code is received and the token is generated.



- Close it.

Slack Connection - Slack ⓘ Folder Add D

Connection OAuth 2.0

Client ID 1549081583281.1545645758612

Client Secret <Encrypted>

Access Token Regenerate 🟢 Reset

- Click Save and close.

Step 23: Click + on operation and name it. Click on import.

Slack Connector Operation Slack Operation ⓘ Folder Add Description


Options Archiving Tracking Caching Import

Connector Action CREATE ▾

Object

Tracking Direction ⓘ Input Documents Output Documents

Error Behavior Return Application Error Responses ⓘ



Step 24: Configure the atom and Click on Next.

Slack Import Wizard

* Required fields.

Browse In Cloud Atom

Atom^ local_atom

Filter ⓘ

Step 25: Click on the chat message and click on Next.

Slack Import Wizard

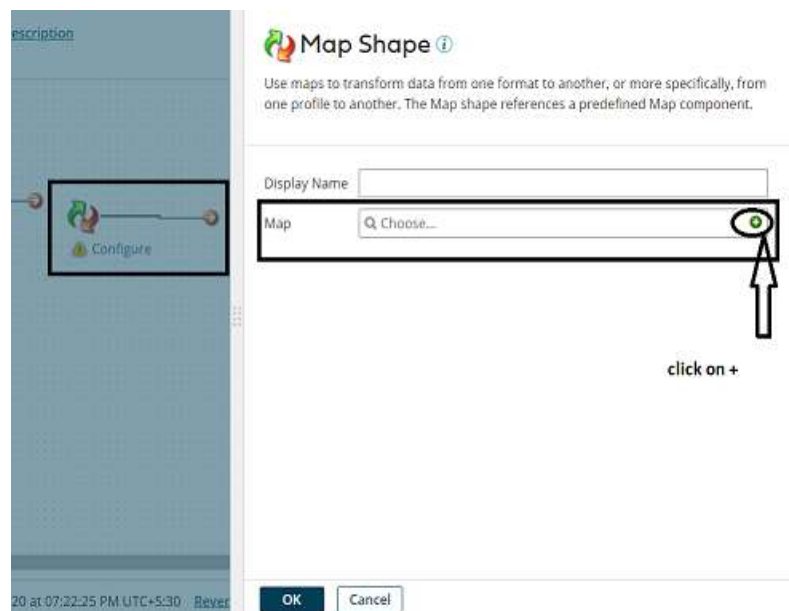


Step 26: We see that profiles are imported. Click on finish, save and close.

Slack Import Wizard

Operation Loaded
Object Name Chat Message
Request Profile Slack Chat Message CREATE Request
Response Profile Slack Chat Message CREATE Response

Step 27: Drag and drop the Map shape onto the process canvas. Click + on the map to create one as shown below.



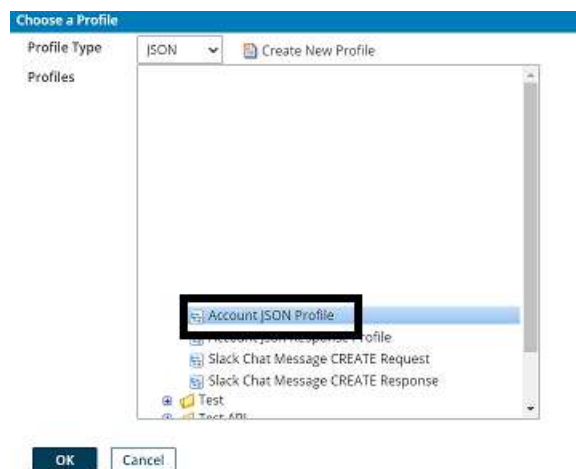
- Name the map and add the profiles by selecting the choose button on both sides. The left side will be the source profile and the right side will be the destination profile.



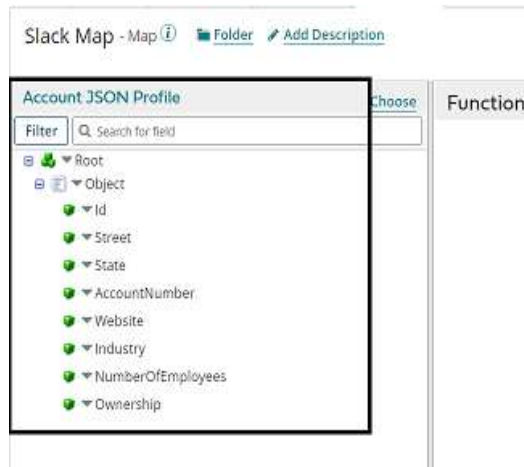
- Now, let us add the profiles. First, we will add the profile at the source side which we have added in Listener Operation.



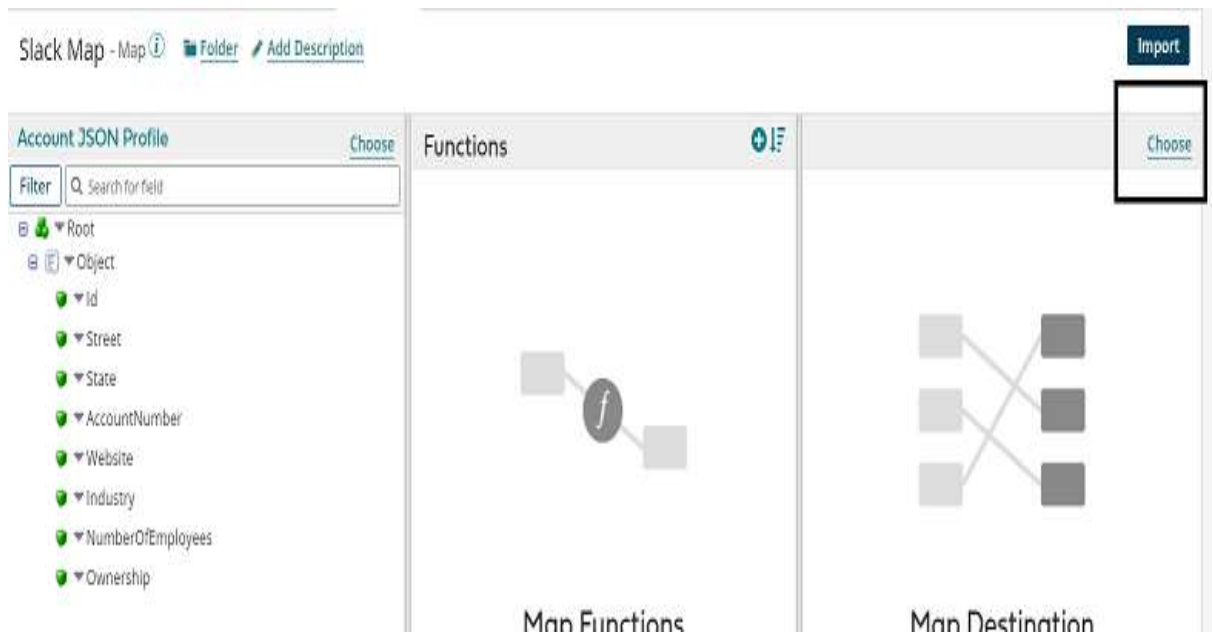
- Select Profile Type as **JSON** and choose the profile.



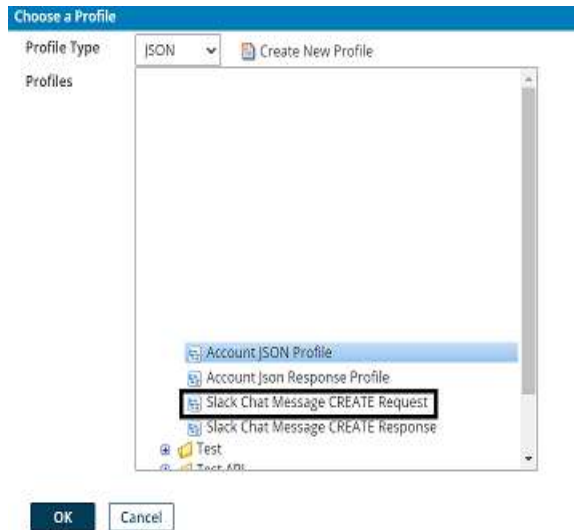
- We see that the JSON profile has been added to the map on the source side.



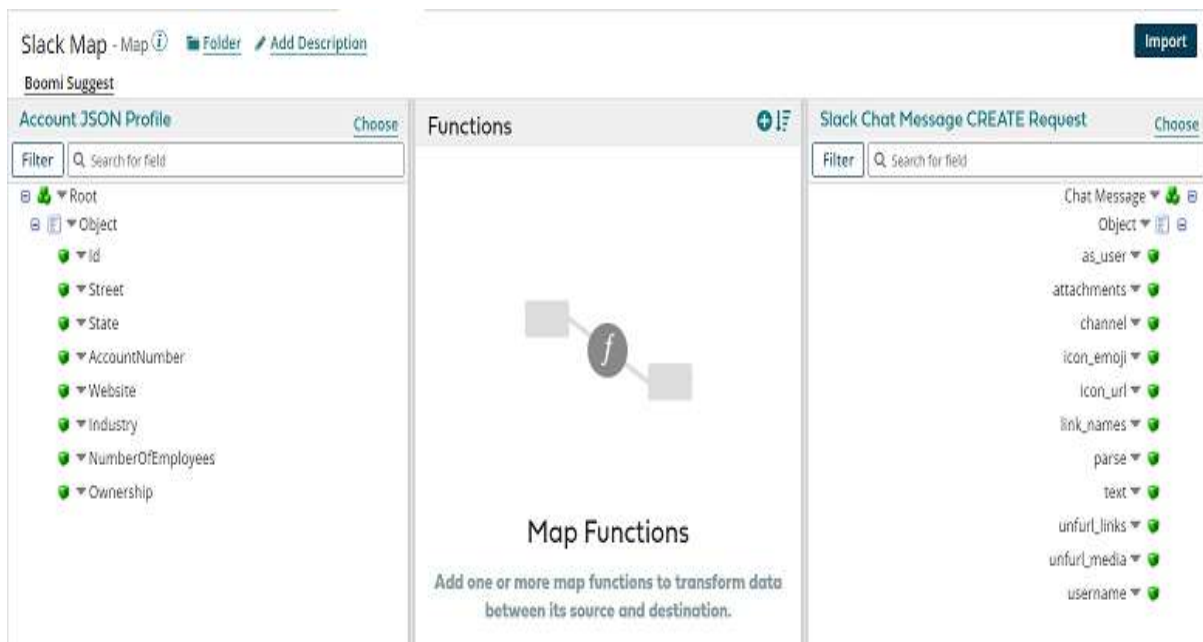
Step 28: Now, we will add the profile at the destination side. In this use case, it would be the Slack Message Chat Request profile which we have imported into Slack. Click on Choose on the right-hand side of the map.



- Choose Profile Type as JSON and select the profile. Click ok.



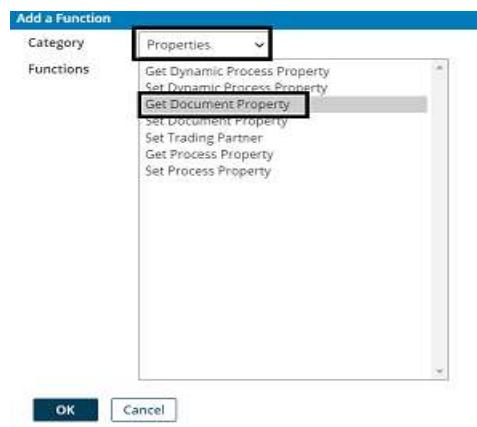
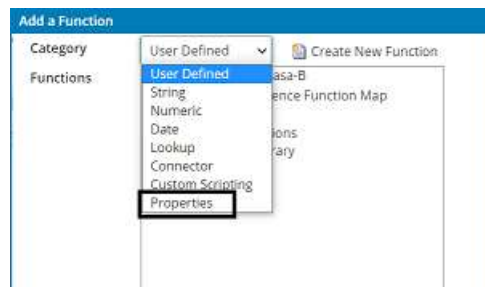
Step 29: We see that the Slack request profile has been added to the map on the Destination side.



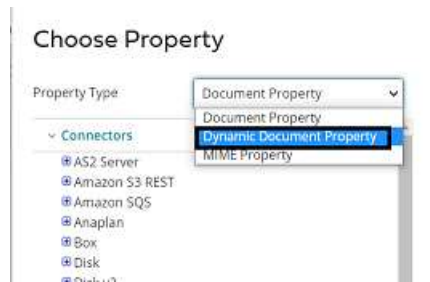
- Now, we will get dynamic document property from functions to get current data and it goes as a body to the attachment. Click +on Functions.



- We can see a few categories in Function i.e. String, Numeric, Date, Look up and so on. Here, we will select Category as properties and function as Get Document Property. Click ok.



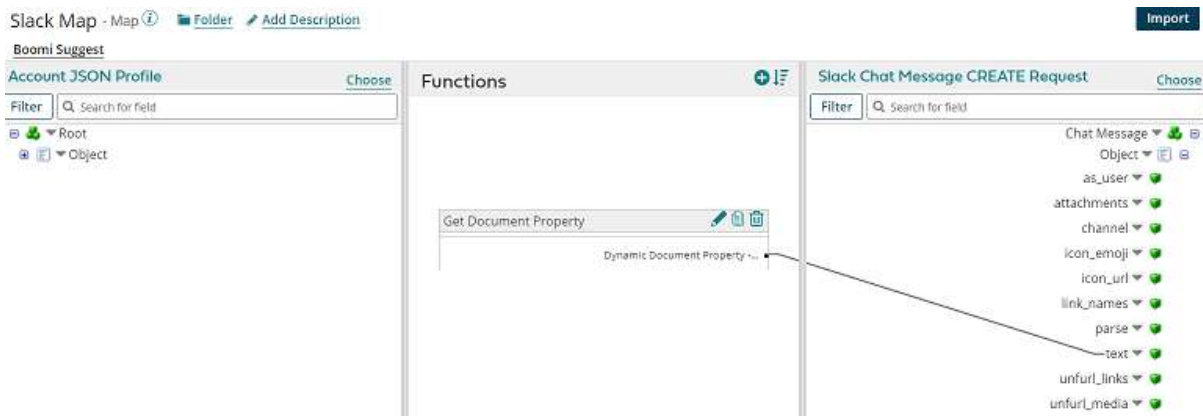
- Choose Dynamic Document Property from the drop-down as shown.



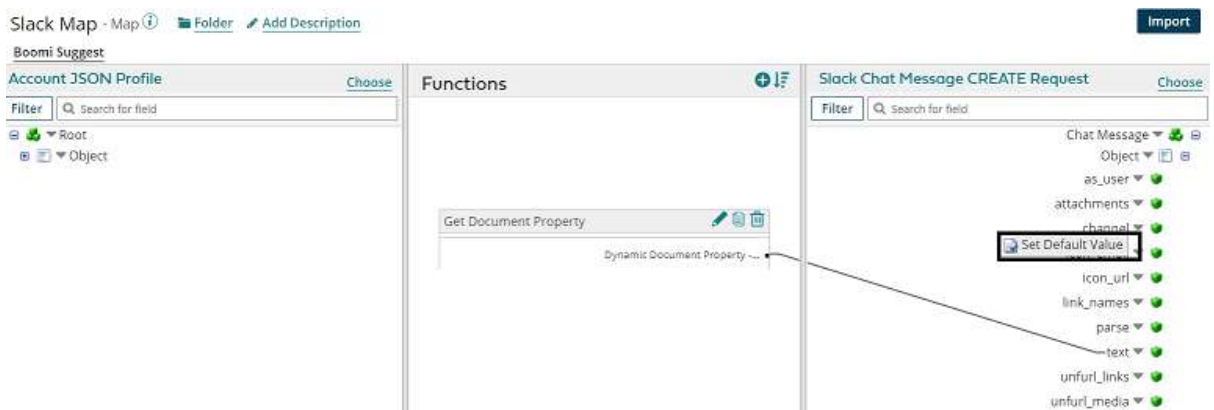
- Here, the Property Type would be Dynamic Document Property and the Property Name would be Text Data which we have set in Set Properties shape. Click ok.



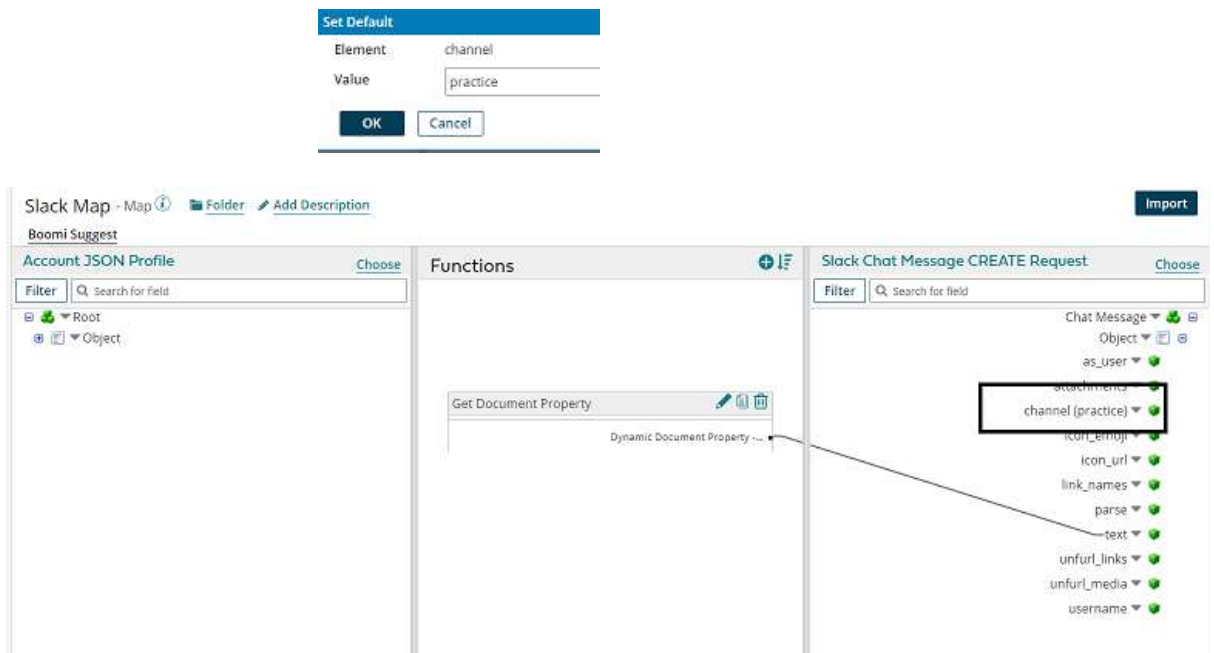
Step 30: We see that we can retrieve the Dynamic document Property in the map. Now, set the property to text as shown in the screenshot.



Step 31: Now, we have to set a default value for the channel. we have already created a channel as practice while creating the account. Click on the channel and set a default value.



- Add value as a practice



Step 32: Drag and drop the return document shape onto the process canvas to return the response which comes from the message body.



Step 33: Arrange all the shapes in an order and deploy the process. Click on Create Packaged Component.



- The process gets selected automatically. Click Add Details.

Create Packaged Components: Select Components

Select one or more deployable components from the Component Explorer to create a packaged component of each selected results in its own individual version to be deployed independently. If the selected component was packaged previously, det.



Next: Add Details

- Next, select the version and write notes if you have any. Click on Create Packaged Component.



Create Packaged Component (1)

- Now, we see that the package has been created successfully and click on deploy.



- We will then have to select the environment. Choose production and click select version and review.

Deploy: Select Environment

Select the environment in which to deploy your packaged component(s), and optionally add notes about the deployment.

Deployment Environment

Deployment Notes

- API Production
- Test
- production

4000 characters remaining.

Deploy: Select Environment

Select the environment in which to deploy your packaged component(s), and optionally add notes about the deployment.

Deployment Environment

Deployment Notes

- API Production
- production**

choose production

4000 characters remaining.

Next: Select Versions

Next: Review

- We will be asked to cross-check the environment which we have configured in the deployment tab.

Deploy: Review

You're almost done! Before deploying the version of your packaged component, confirm that the destination environment you have selected is correct.

Environment: production
Deployment Notes:

Name	Type	Selected Version	Deployed Version	Duplicate
Slack Test	Process	1.0	N/A	—

- Once, we click on deploy we will be able to see that deployment is done successfully. Close it.

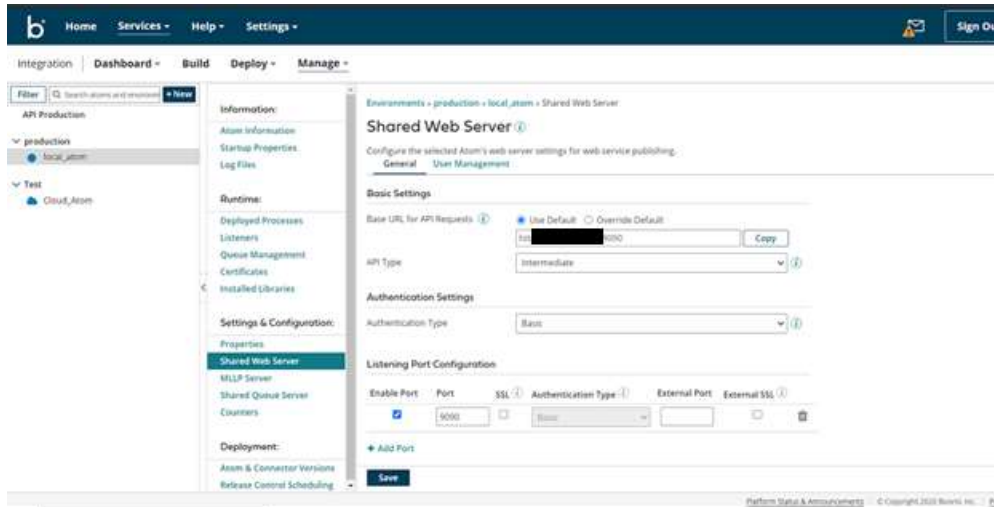
Deployment Successful

Your packaged components were successfully deployed.
Click on the View Deployments button to see all deployments for this account.

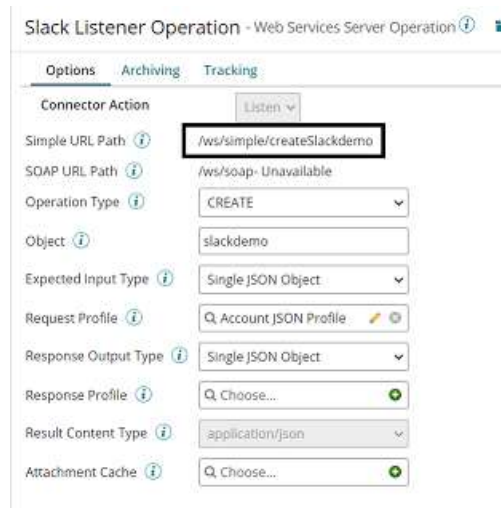
View Deployments

Close

Step 34. Go to Manage > Atom management Click on the atom to which we have deployed the process and choose shared web server. Then, copy the URL and append it with the rest path which we have configured in the operation



Step 35: Go to the Build page click on Slack Listener Operation and copy the rest path.



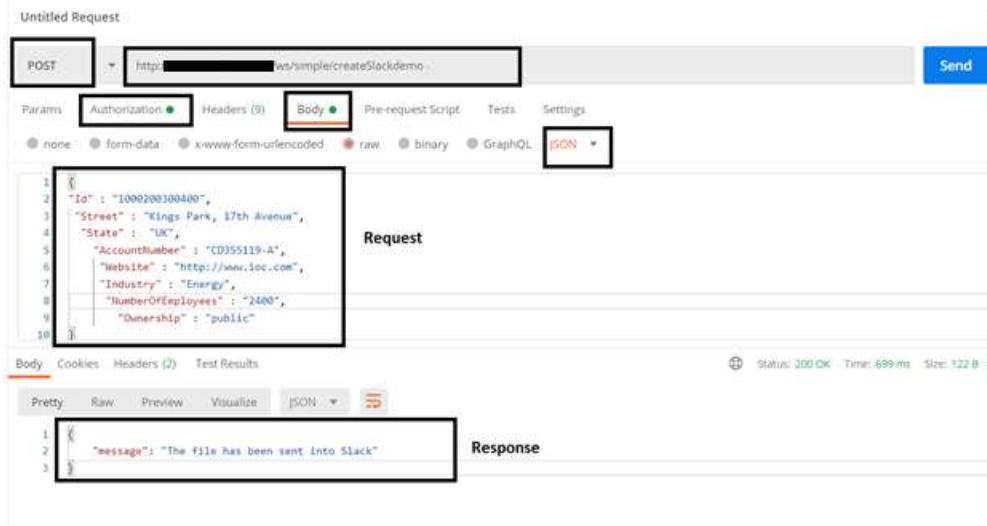
Step 36: Append Base URI with simple URI Path which indeed looks like,

http://[redacted]/ws/simple/createSlackdemo

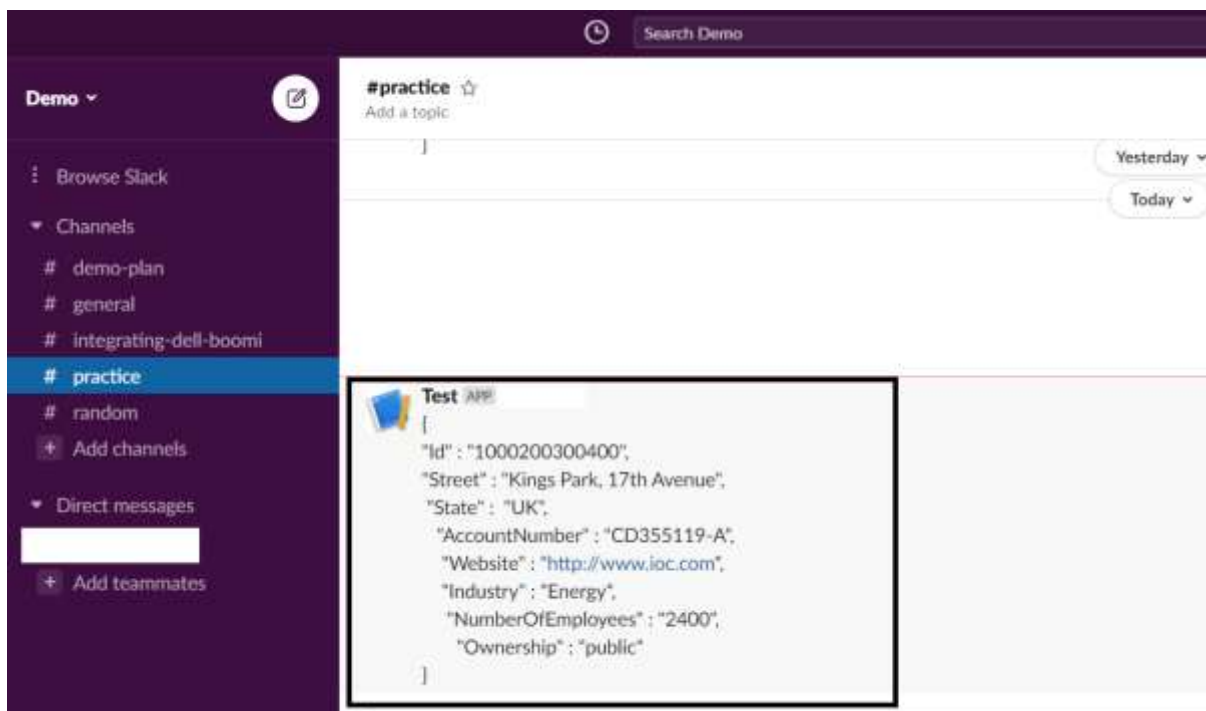
Step 37: Now, test the web service in Postman by giving the following details.

- Give Operation as POST.
- Set the complete URL in the URL Section.
- Get the Authorization Details by navigating to Manage > Atom Management > choose the atom > Shared web server > User Management and get username and token and give it in Authorization.

- In Body, select raw followed by JSON and give the Json request profile which we have added in the listener request profile and add the content
- Click on send.



Step 38: Go to Slack and we see that the JSON file has been sent as a message in the practice channel.





TGH

Making Integrations Simpler



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

