Integrating with Salesforce using a REST API created in IBM Application Integration Suite 1.0

SanjayNagchowdhury
Published on March 18, 2016 / Updated on September 23, 2016

In this article, I describe how to expose CRUD operations on Salesforce objects using a newly defined REST API in IBM Integration Bus by using new capability delivered in IBM Application Integration Suite 1.0. This fixpack contains many new features which include:

> Enhancements to the REST API Editor so that a REST API can be created from scratch.
>
> Enhancements to the Graphical Data Mapping capability so that model definitions can come from a Swagger document or a JSON schema.
>
> A Salesforce Request node which allows connectivity with Salesforce so that records can be created, retrieved, updated or deleted for Salesforce objects such as Accounts, Contacts and Leads.

Salesforce is a leading CRM software and enterprise cloud ecosystem. By using the Salesforce Request node, you can integrate your business critical applications in your organization with Salesforce. For example, you can synchronize your master list of customers, products, prices and other business critical data with your key applications. Salesforce has many different type of objects such as Accounts, Contacts, Leads and Opportunities. By using IBM Integration Bus, you can develop message flows to interact with these Salesforce objects and with a host of other service endpoints such as SAP and Oracle.

This article will show how these three new features can be used together to receive message data, transform it and send it to Salesforce. I will describe the steps to define a REST API which allows you to create, retrieve, update and delete a Salesforce Contact. The scenario will show how to quickly create a new REST API, transform the message that is received using the graphical mapping capability and send the message to Salesforce using the Salesforce Request node.

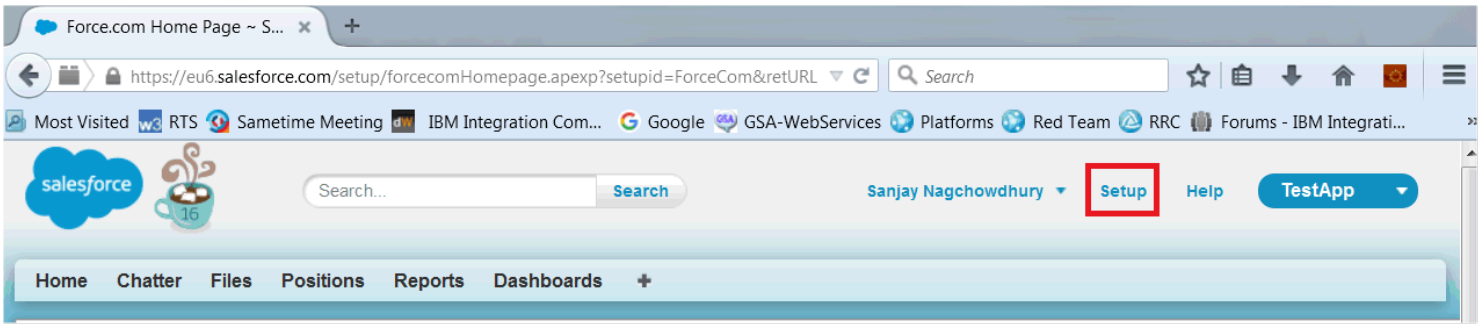
If you want to use the Salesforce Request node you need to purchase IBM Integration Bus as part of the IBM Application Integration Suite bundle. You can use the Salesforce Request node in `developer` mode while you are developing and testing your flows, but you cannot deploy the flow into a production environment until you have modified the integration node to run in `applicationIntegrationSuite` mode. You do this by running mqsimode:

```
mqsimode integrationNodeName –o applicationIntegrationSuite
```

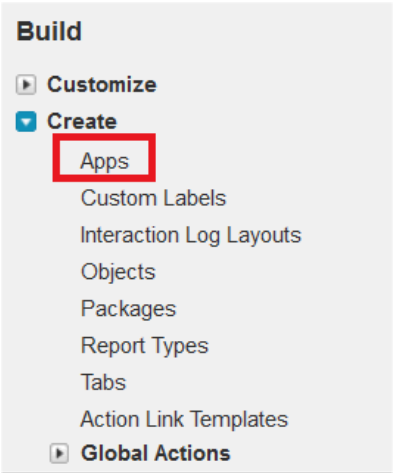
Set up Salesforce security credentials

In order to connect to Salesforce from IBM Integration Bus, you need to create a Salesforce Connected App.

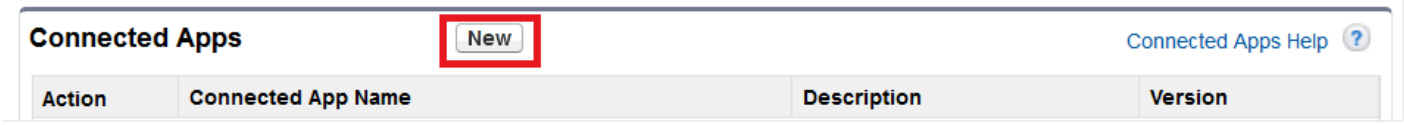
Log onto Salesforce.com and click on Setup



On left-hand side, click on Build->Create->Apps:



Scroll down the page. In the Connected Apps section, click on New:



Fill in the following:

Connected App Name (you can call it IBM Integration Bus)

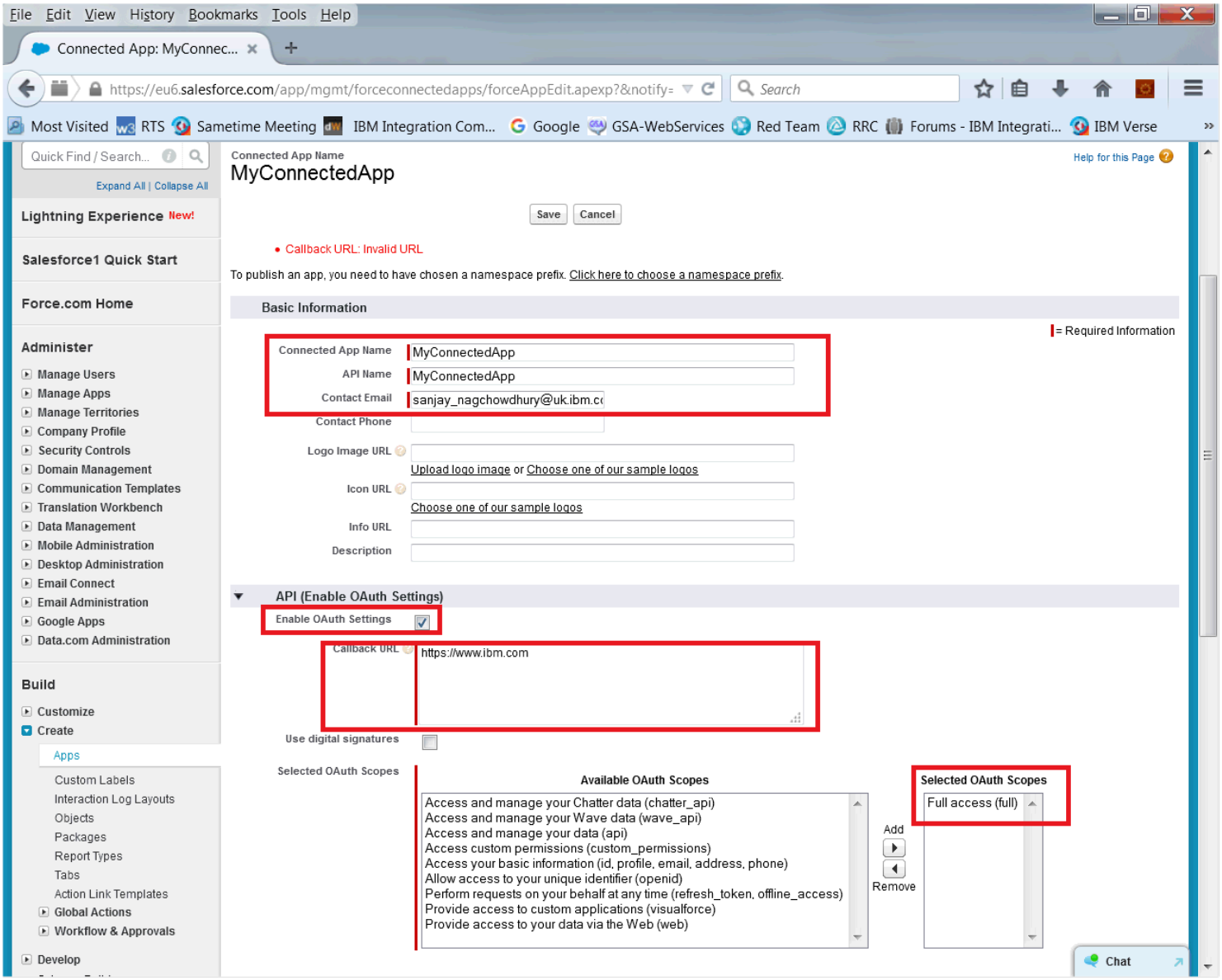
API Name

Contact Email

Callback URL (Set the Callback URL to any valid secure URL. The URL will not be used by your IBM Integration Bus Connected App).

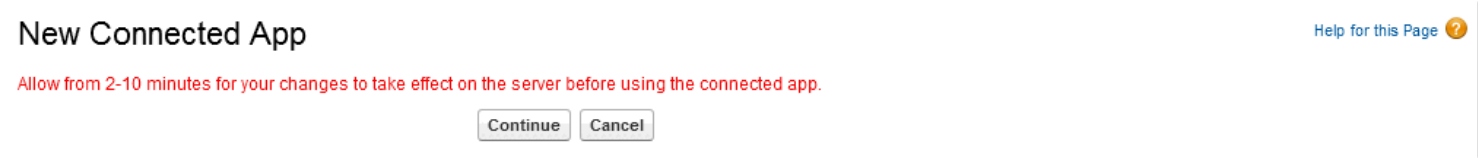
Select Enable OAuth Settings

Choose Full access (full) from Available OAuth Scopes



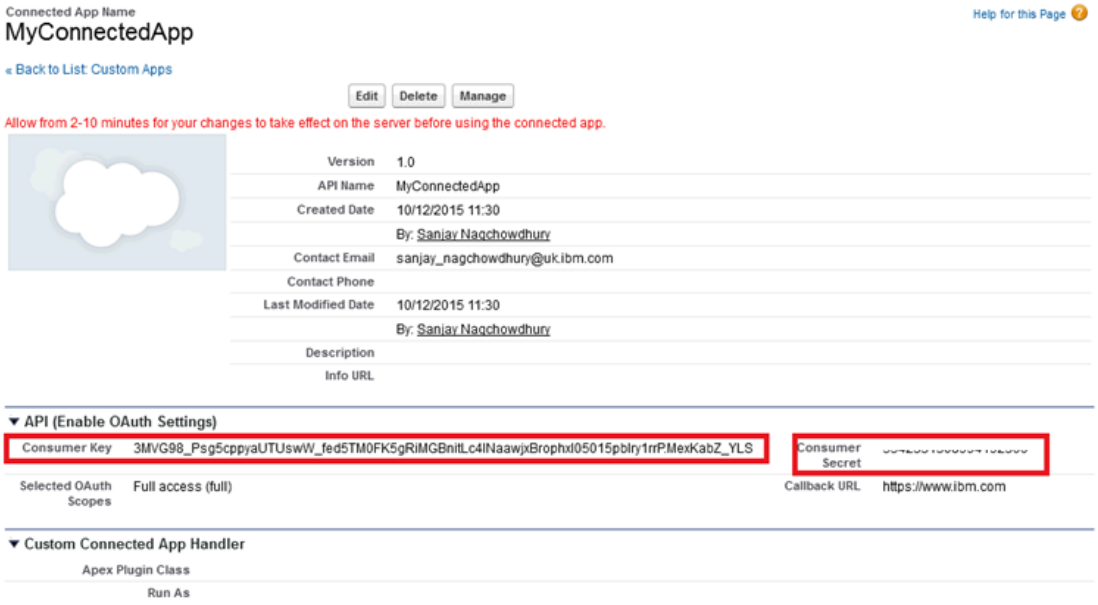
Click Save:

You see the following. Click on Continue:



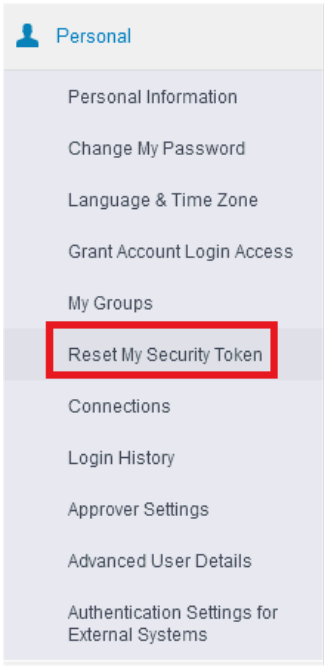
The new connected app will be shown. You need to take a note of the Consumer Key and Consumer Secret. These are needed by IBM Integration Bus to connect to Salesforce:

Now, you need to reset your security token. This is also needed for connecting to

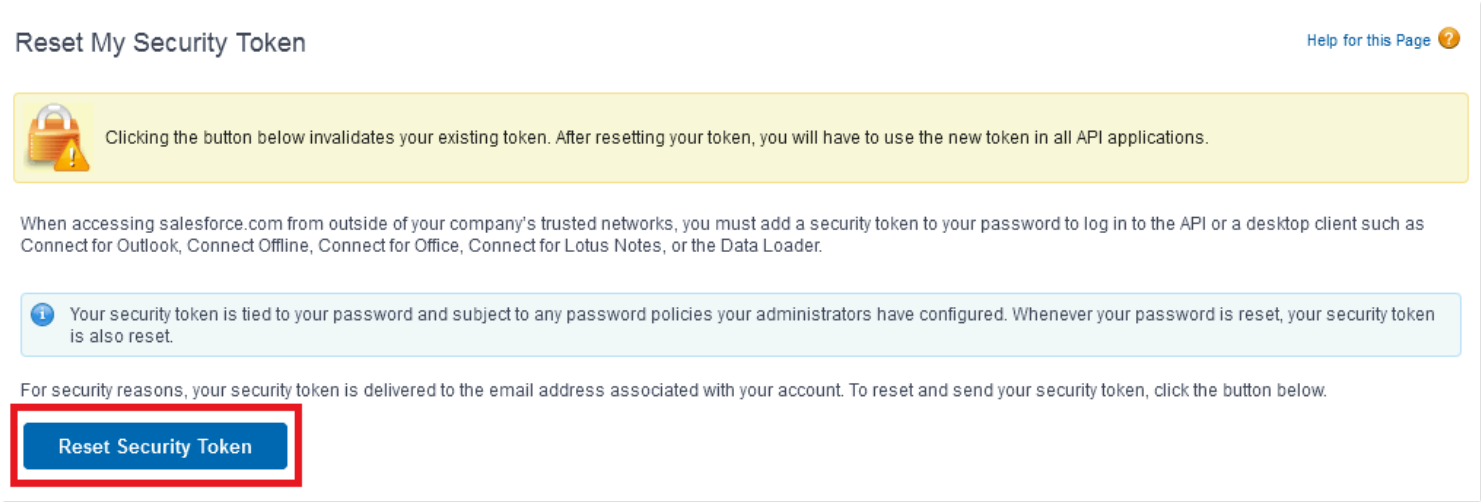


Salesforce. To do this, click on your name and then ‘My Settings’:

Click on ‘Reset My Security Token’



Click on the ‘Reset Security Token’ button:



The security token will have been sent to your email address. Now, you have the information needed to connect to Salesforce:

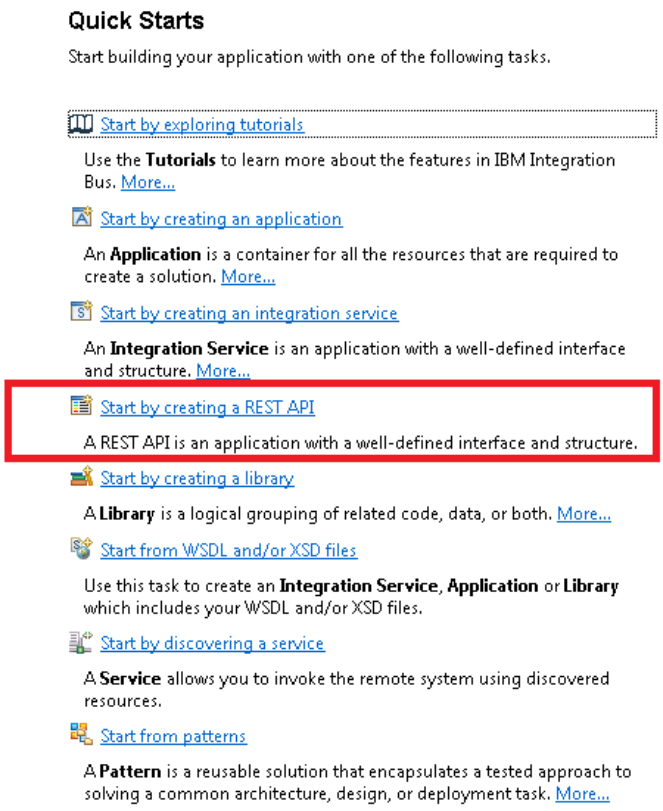
- Salesforce Userid
- Salesforce Password
- Security Token
- Consumer Key
- Consumer Secret

You need to create a security identity for the Salesforce credentials which are stored using mqsisetdbparms. The password must be suffixed with the security token that was supplied to you by Salesforce. You should create the security identity by running mqsisetdbparms like this:

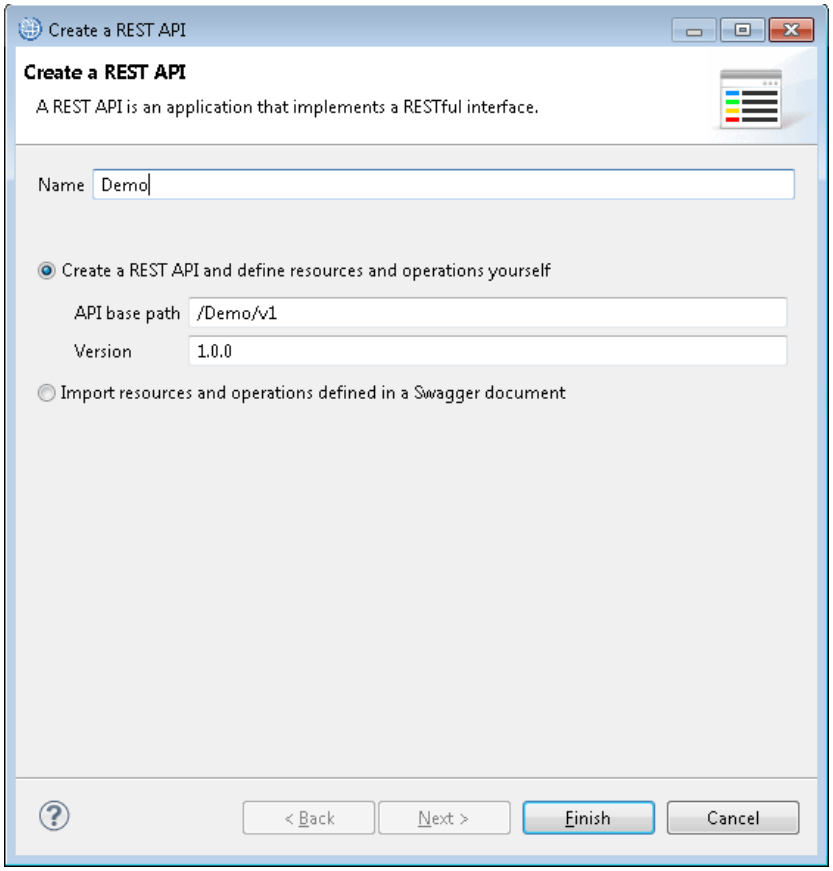
```
mqsisetdbparms integrationNodeName -n salesforce::mySecurityIdentity
-u <Salesforce Userid> -p <Salesforce Password concatenated with
Security Token>  -c <Consumer Key> -s <Consumer Secret>
```

Create a REST API

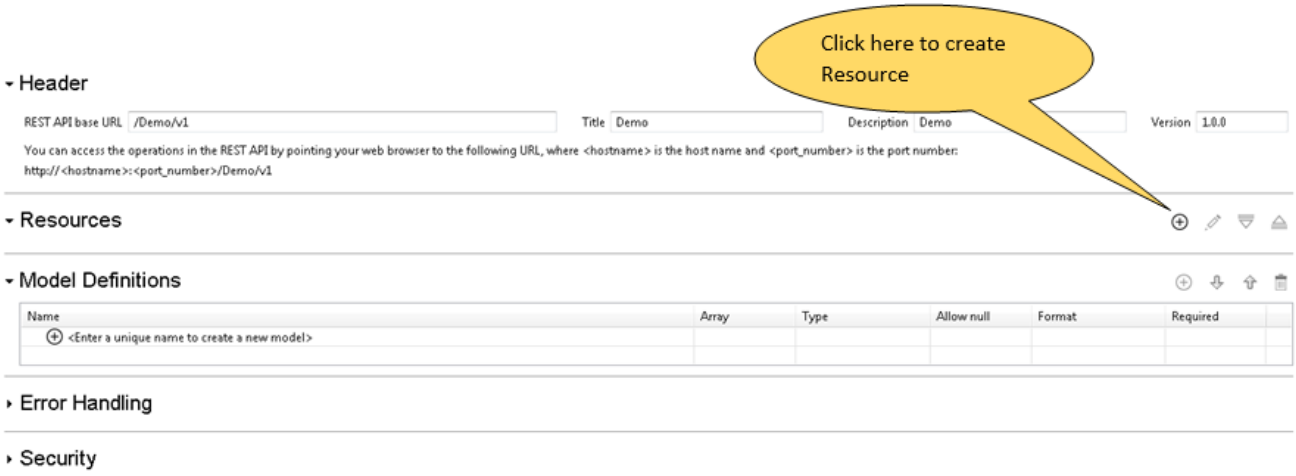
Now that you have your security credentials set up, you can create a REST API to expose CRUD operations on Salesforce objects. There is a link from Quick Starts in the toolkit, or you can select File -> New -> REST API



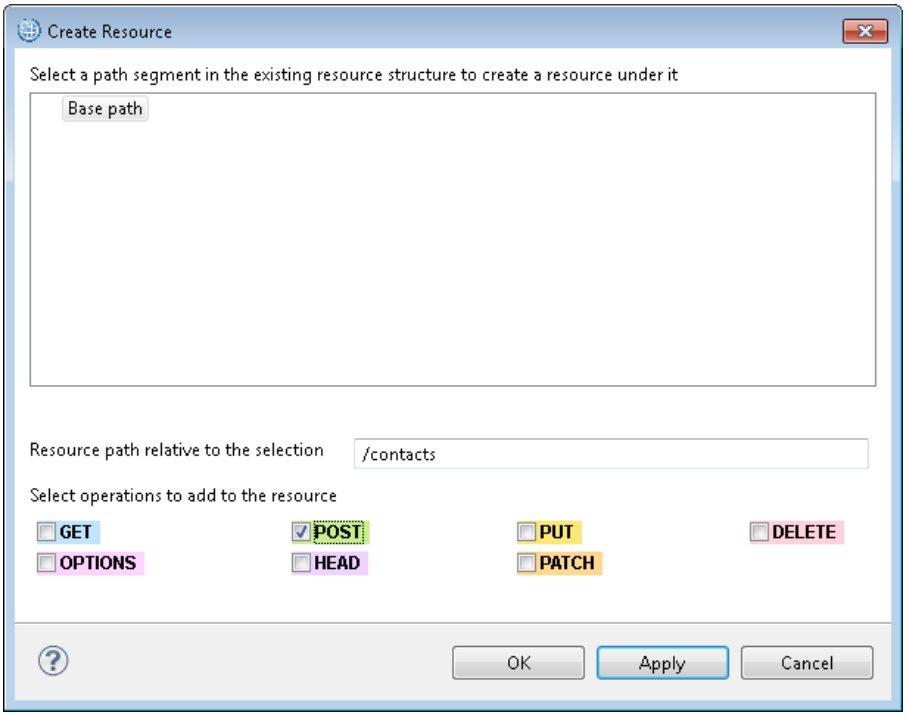
You enter the name of your REST API which forms part of the API base path:



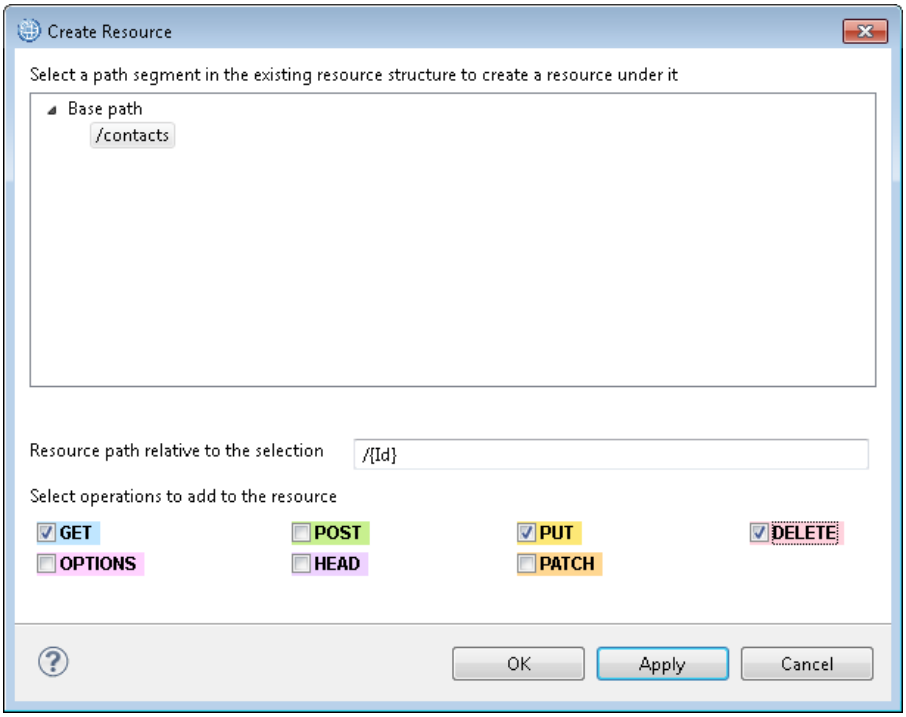
After clicking Finish, you see the REST API Editor. The first thing to do is to create a new Resource:



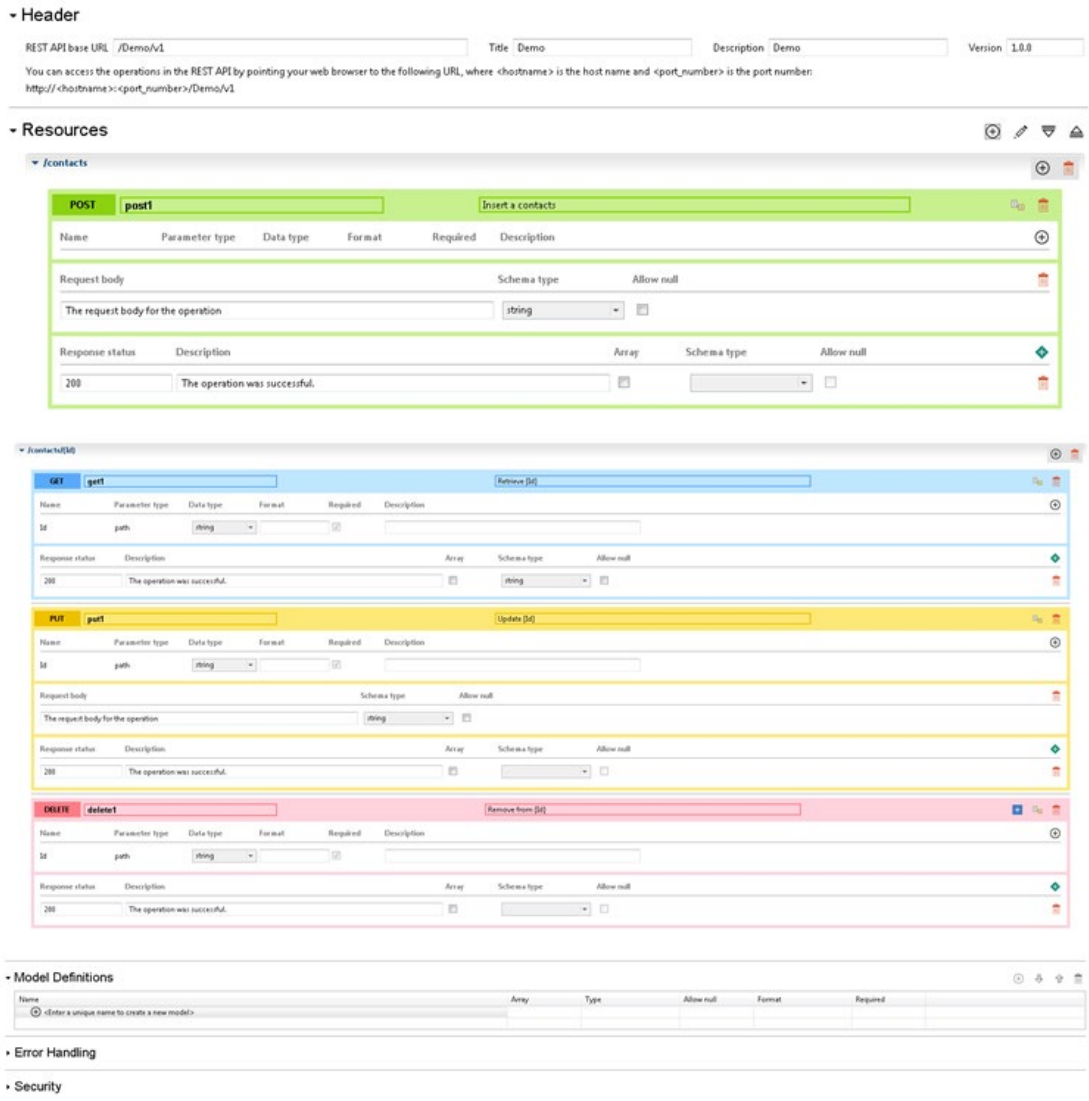
You are shown a window to select the operations for the Resource. Specify a POST operation for a Resource called /contacts so that you can create a Contact in Salesforce.



Specify another Resource called /contacts/{Id} so that you can retrieve a contact by Id (GET), delete a contact by Id (DELETE) and update a contact by Id (PUT).

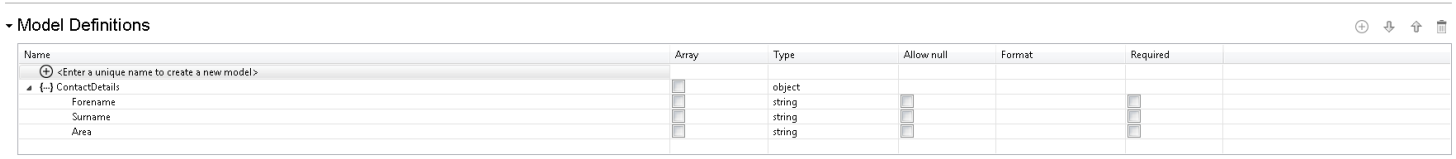


After creating these Resources, you see this in the editor:

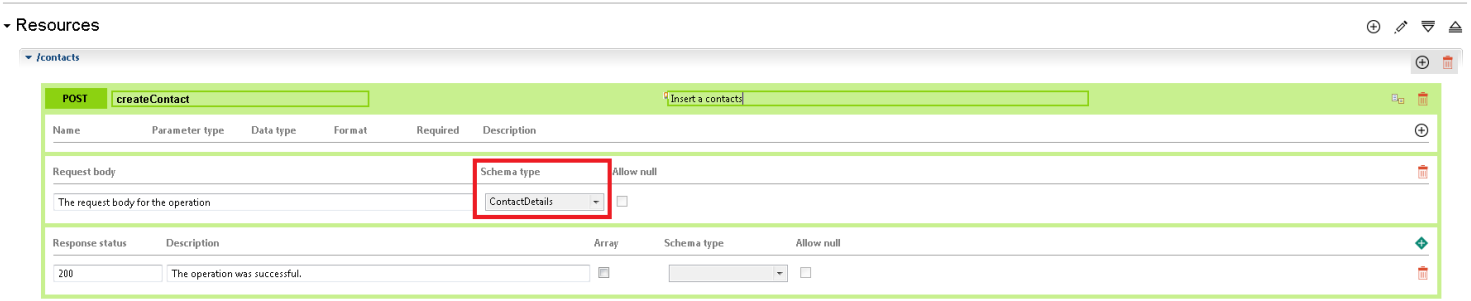


Each operation will be implemented by a subflow. You can change post1, get1, put1 and delete1 to use more meaningful names which will form the name of the subflow. For example, you could change them to createContact, getContactById, updateContactById and deleteContactById.

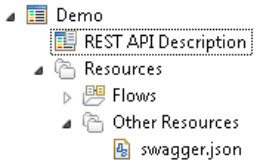
Next, you create a new model definition which is used by the Request and Response body for each REST API operation. Define a model called ContactDetails which is a JSON object and contains three children: Forename, Surname and Area, each of which are JSON strings as follows:



The createContact POST operation will use ContactDetails as the schema type used by the Request Body.

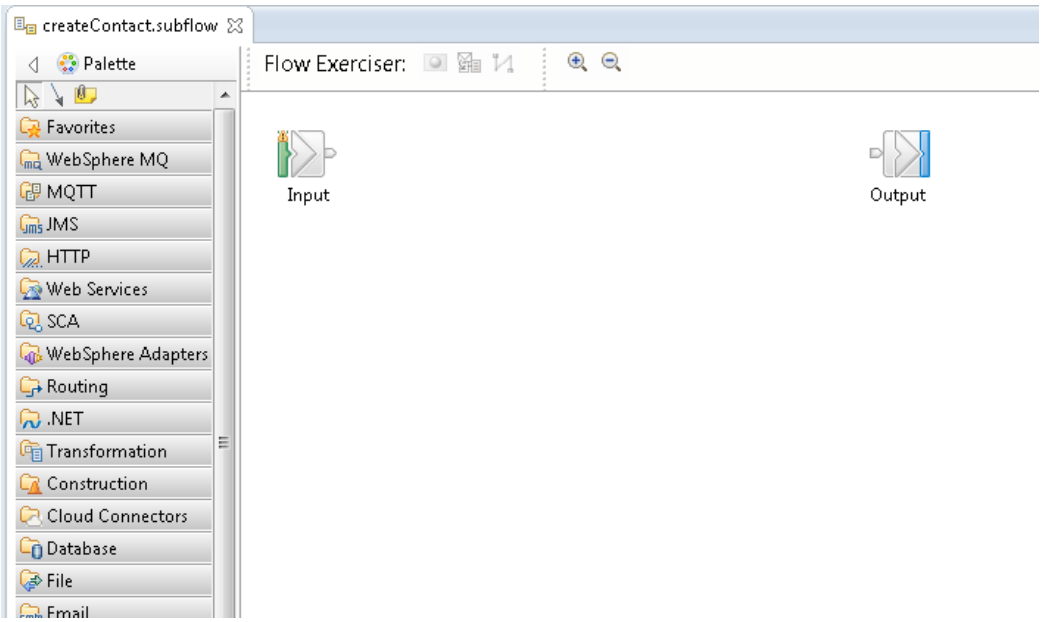


If you look in the Application Development tab, you will see that a swagger.json has been generated and it is located under Resources/Other Resources:



Now you can implement the createContact POST operation. To do this, click on the Create a subflow button:

After clicking on the button, you see the Message Flow Editor is created for the createContact subflow:

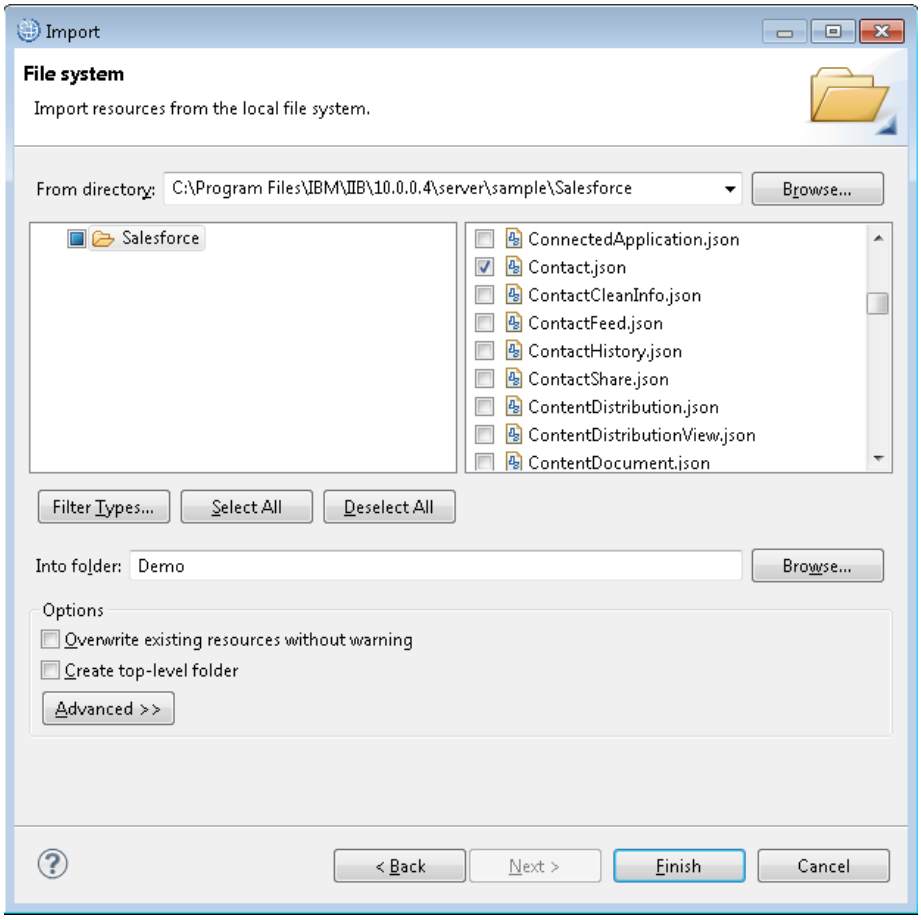


Define a mapping for the Salesforce Contact object

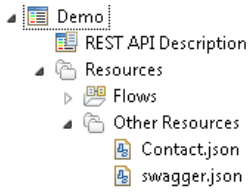
Now you need to add a mapping node to the subflow so that the flow can transform the REST request message that is received to a format that is understood by Salesforce.

In IBM Integration Bus v10.0.0.4, sample JSON schemas have been provided for Salesforce objects in install_dir/server/sample/Salesforce directory, where install_dir is the directory in which you installed IBM Integration Bus. The relevant schema can be imported into your application or shared library so that you can use it as the input or output message map in a Mapping node.

Since you are going to interact with a Salesforce Contact, you need to import the schema for it which is called Contact.json.

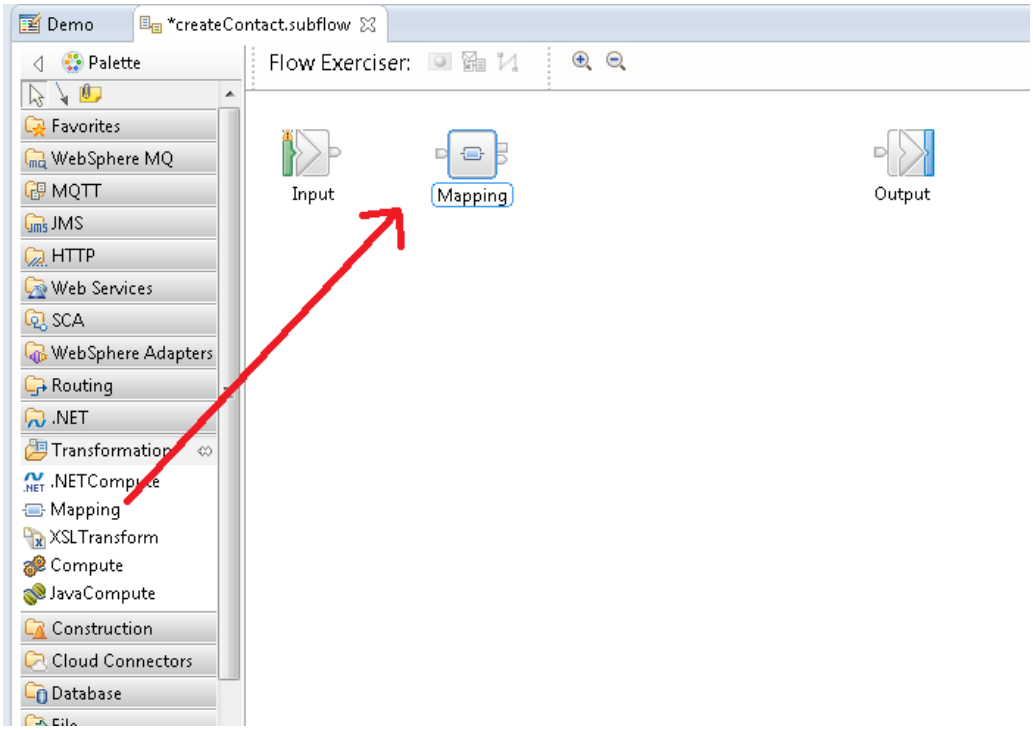


After importing it from the file system, you will see that Contact.json is also listed under Resources/Other Resources:

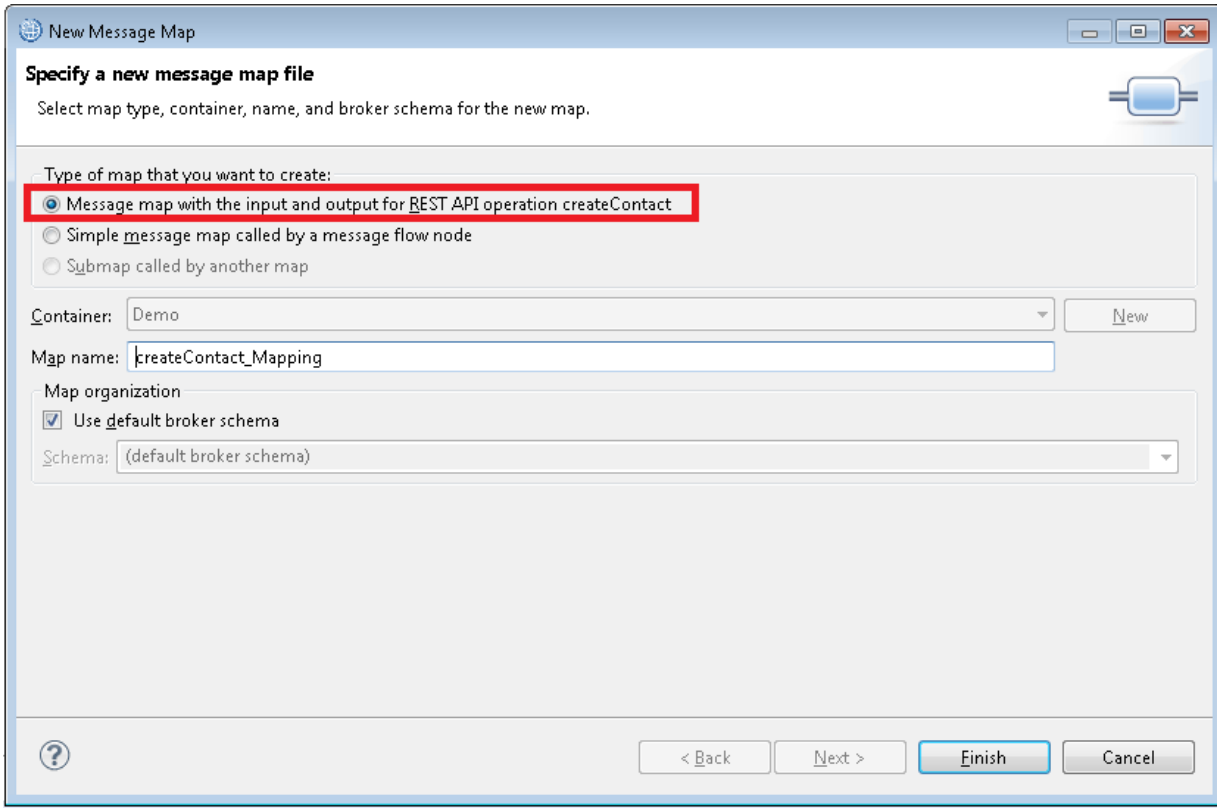


The JSON schema must reside in either a REST API or in a shared library.

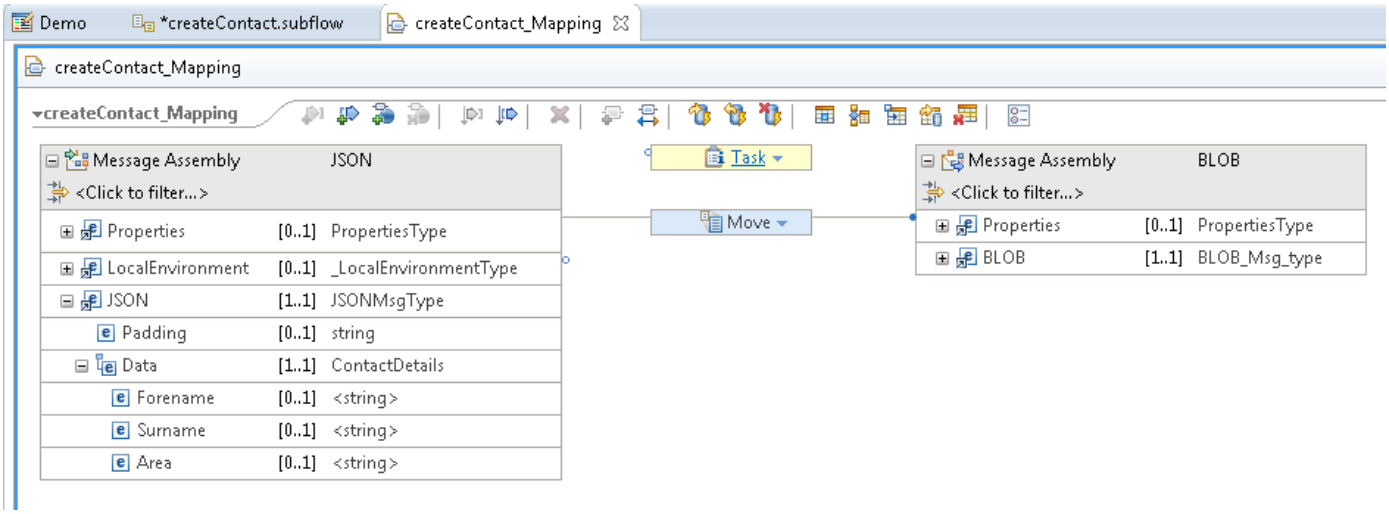
You are going to use the swagger.json for the input message map and the Contact.json as the output message map in the map used by the Mapping node. Select the Mapping node from the Transformation drawer and drag and drop it onto the flow editor:



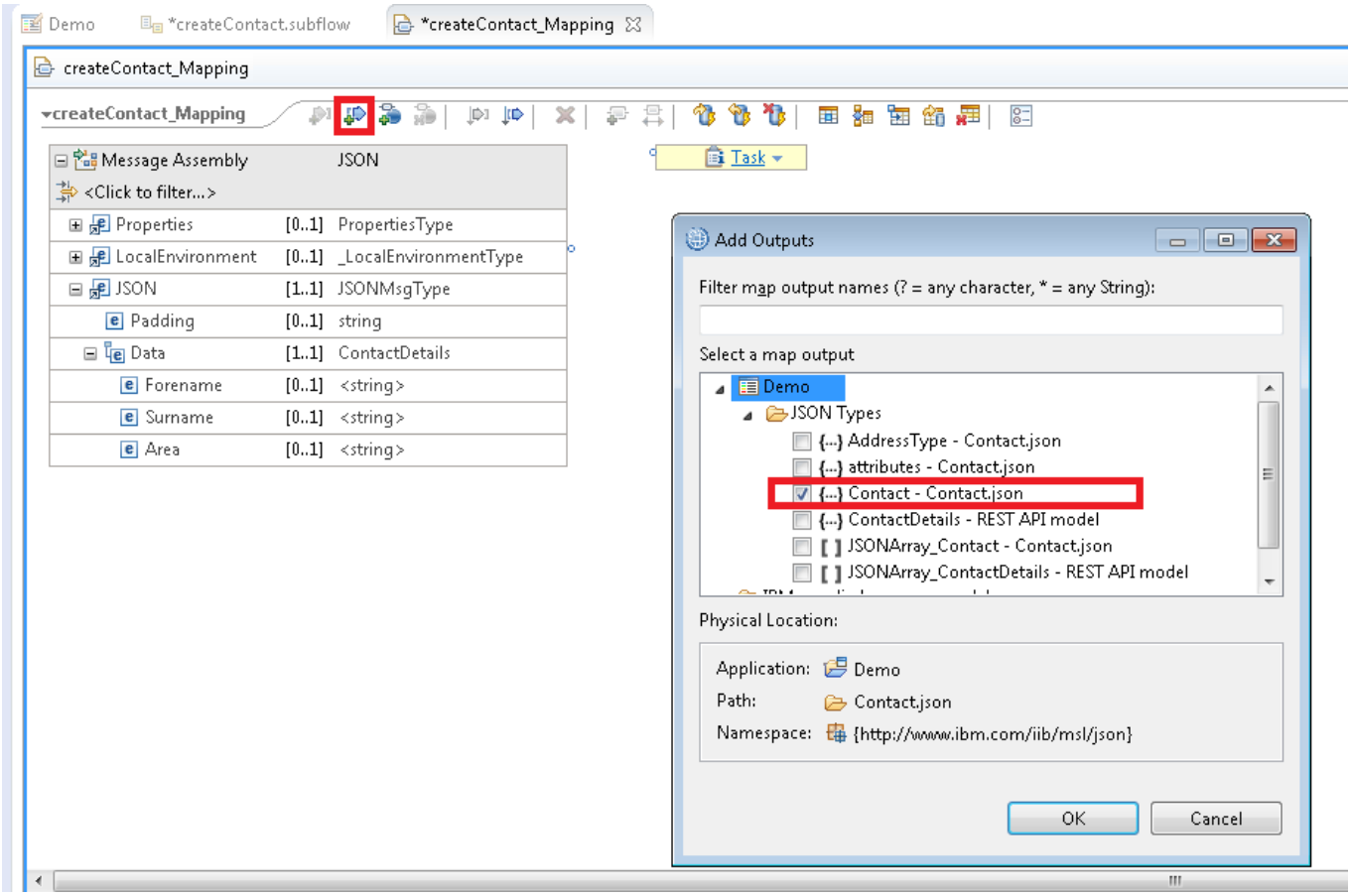
To configure the mapping node, double click on it. You will see a dialog showing that you can map the input and output for a REST API operation. This is a new option that has been provided in 10.0.0.4:



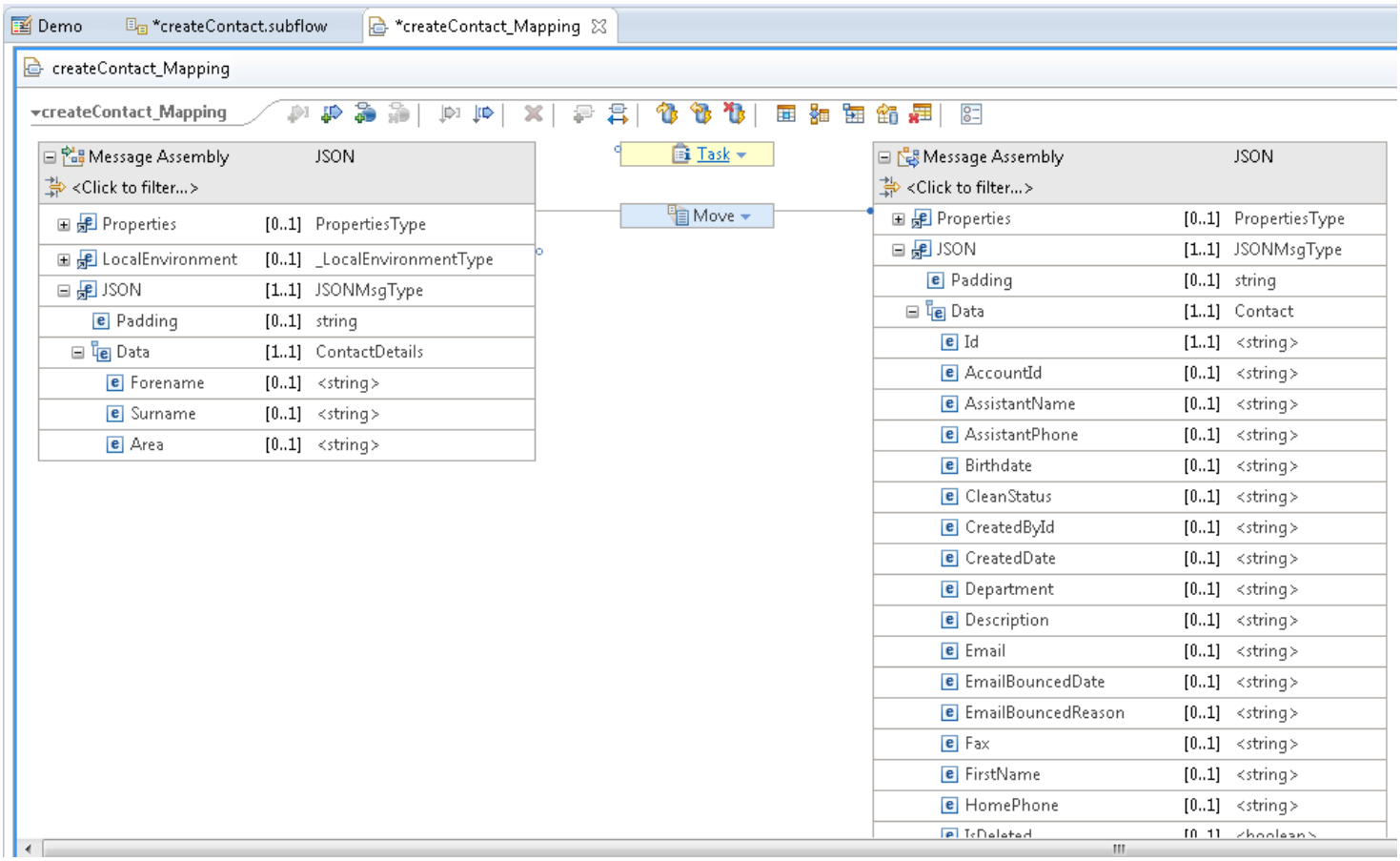
After clicking Finish, you see the initial map that is created. If you expand JSON and then Data on the Input map you can see the ContactDetails data type that you defined. The output message body for the operation is currently BLOB as the operation had no response body data defined for it.



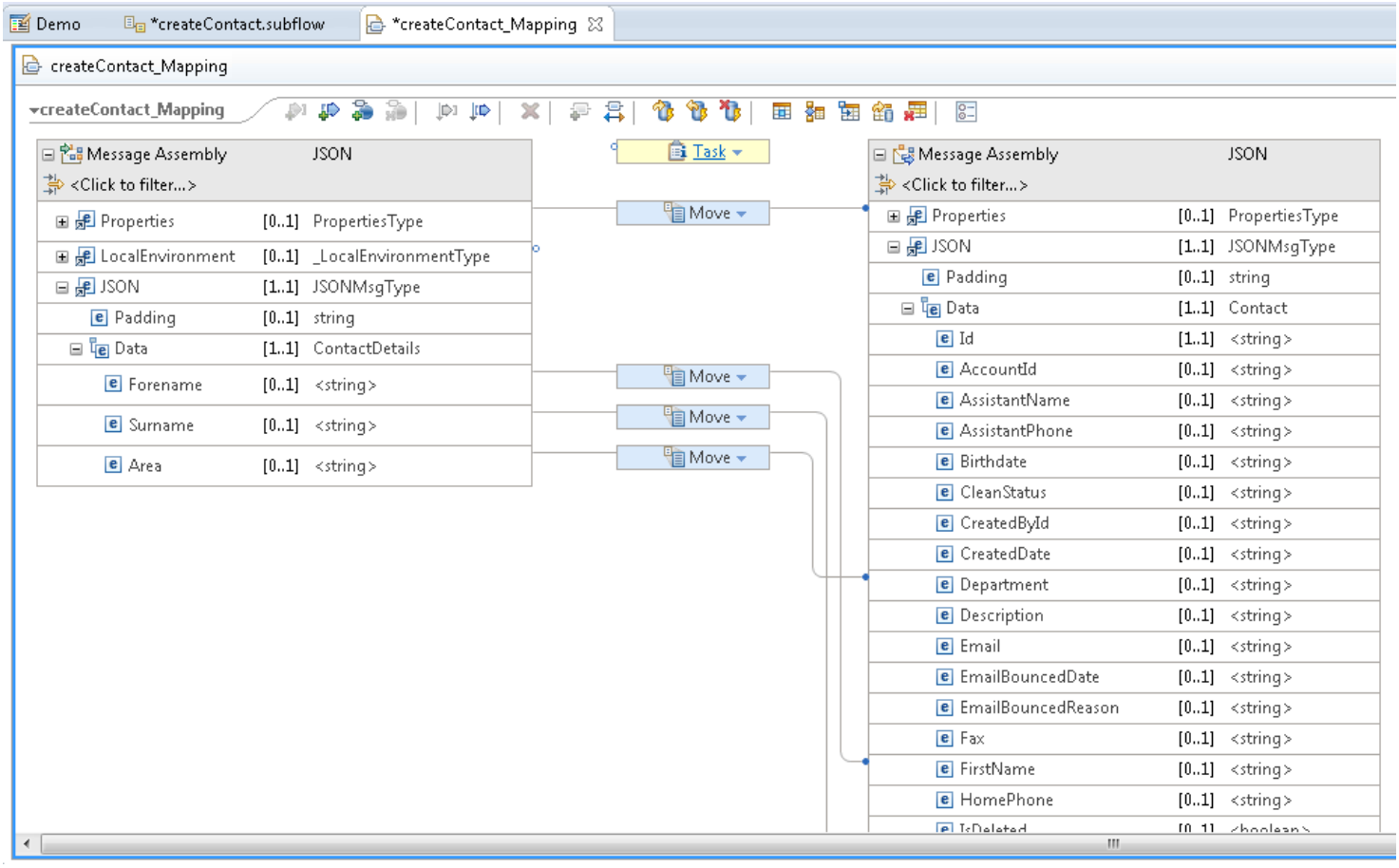
The map is created with the output for directly building the REST API response, however in our case we first need to invoke the salesforce request node, so we will adjust the output to provide the JSON data required. You need to delete the output message assembly that was created and replace it with the message map for Contact.json. Simply click on the output Message Assembly and click Delete. Now click on the “Add an output object” button, select Contact.json and click OK.



After selecting Contact.json, you will see the Message Assembly for the output map:



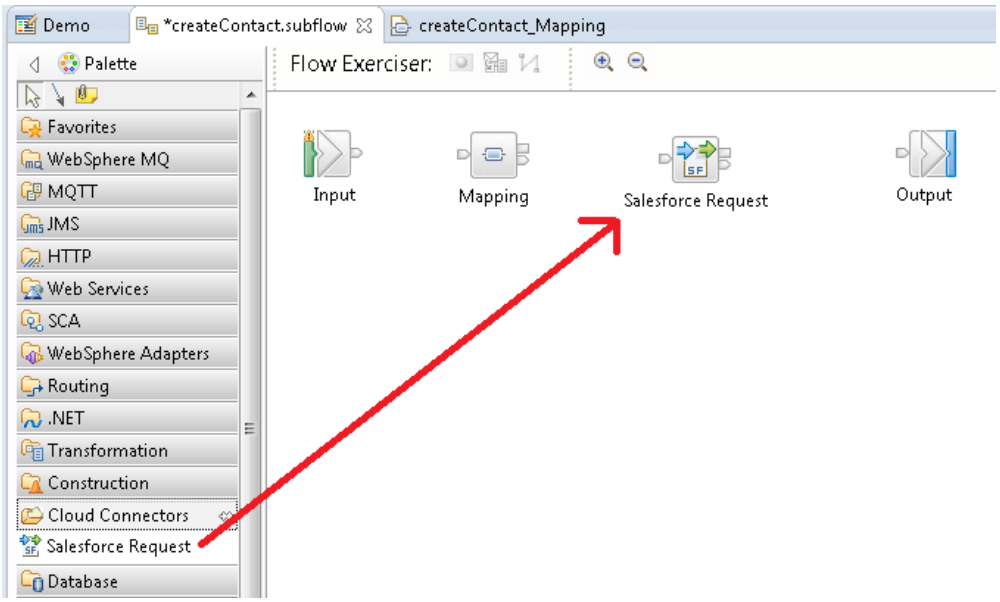
Now you can map Forename to FirstName, Surname to LastName and Area to Department. You do not need to assign values to all the fields as they are optional.



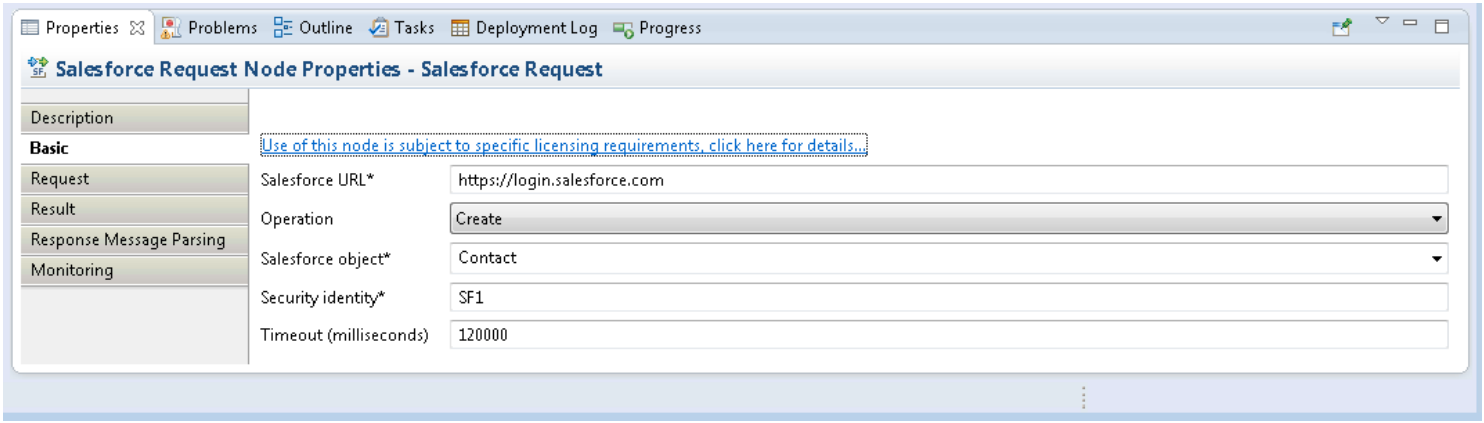
You have now defined the transformation for the REST API request message to be transformed into a format which is compatible with Salesforce.

Configure the Salesforce Request node

In IBM Integration Bus 10.0.0.4, there is a new drawer called Cloud Connectors which contains the Salesforce Request node. Drag and drop the Salesforce Request node onto the subflow.



Configure the details for the Salesforce Request as follows:



The Salesforce URL is the url of the external Salesforce system to which you are connecting.

The Operation can be one of Create, Retrieve, Update or Delete.

The Salesforce object can be chosen from a list of objects or a custom object name can be specified. Here, we choose Contact.

The security identity is the value that was used when storing the Salesforce credentials using mqsisetdbparms. Here we have specified SF1.

The timeout value is in milliseconds. This is the time that Salesforce Request node waits for a response to be received from Salesforce.

It is possible to override node properties so that they are set dynamically while the flow is running. The values are set under the LocalEnvironment.Destination.Salesforce.Request subtree.

You have now implemented the createContact POST operation in the subflow. The subflow does the following:

Receives the REST request for the createContact POST operation

Transforms the message so that it is in the correct format for Salesforce to process.

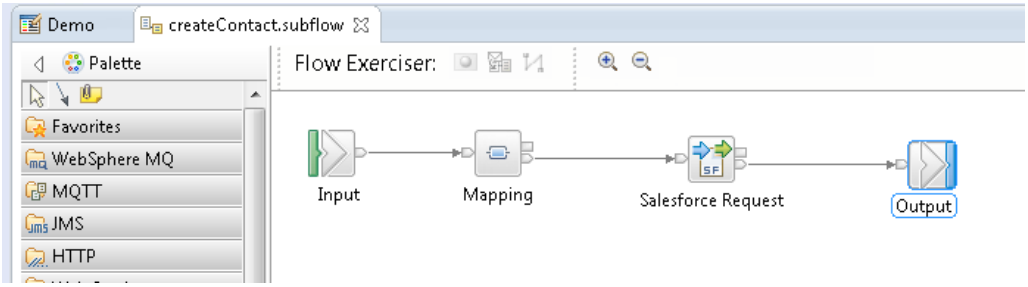
Sends the message to Salesforce.

Receives a response message from Salesforce. The response message contains the same data as the input message that was sent to

Salesforce, augmented with the Salesforce ID for that newly created object.

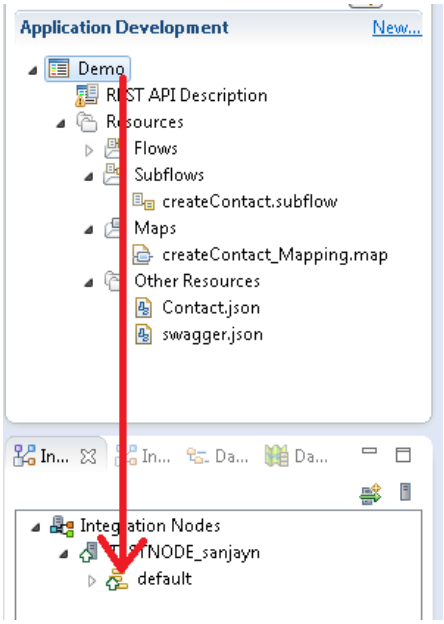
Returns the response message to the originating client who sent the createContact request.

The subflow has done this by just adding a Mapping node and Salesforce Request node to it. The subflow looks like this after it has been wired up:

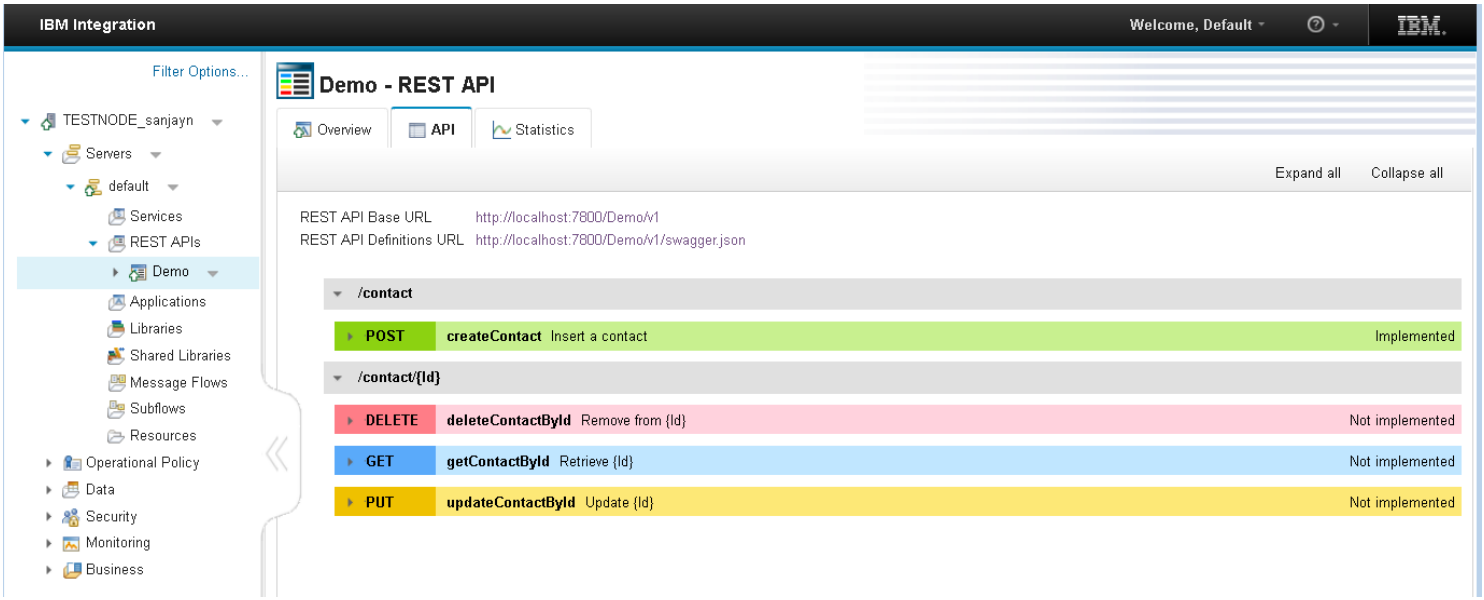


Deploy the REST API and send a message

Deploy the REST API by simply selecting it in the toolkit and dragging and dropping it onto your Integration Server:

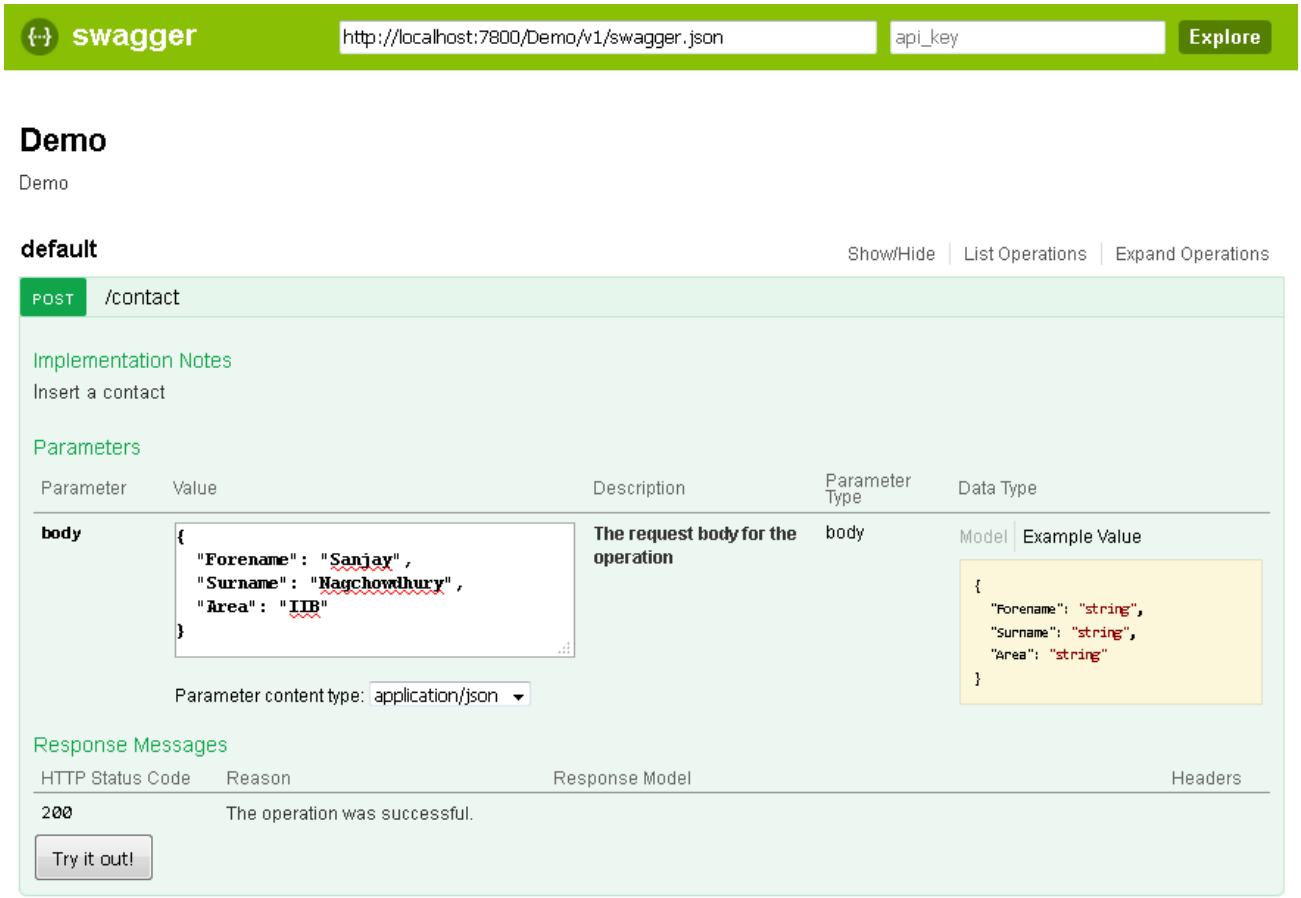


After deploying the REST API, you can administer the REST API in the IBM Integration Bus Web User Interface. By clicking on the REST API and then selecting the API tab, you can see details for the REST API. It shows that the createContact operation has been implemented but the other operations have not been implemented yet. It also provides a link to the swagger.json file which was generated automatically for the REST API.



If you copy the REST API Definitions URL http://localhost:7800/Demo/v1/swagger.json and paste it into a swagger.io, you can send in a request message. You will need to configure the HTTP listeners for your integration server so that they respond to and permit cross-origin requests from a web browser. This can be done by running this command:

```
masichangeproperties integrationNodeName -e integrationServerName -o HTTPConnector -n corsEnabled -v true
```



After submitting the request message, you should see the operation was successful and an Id is returned for the new Salesforce contact that was added.

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	The operation was successful.		

Try it out! Hide Response

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "Forename": "Sanjay",
  "Surname": "Nagchowdhury",
  "Area": "IIB"
}' 'http://localhost:7800/Demo/v1/contact'
```

Request URL

```
http://localhost:7800/Demo/v1/contact
```

Response Body

```
{
  "Department": "IIB",
  "FirstName": "Sanjay",
  "LastName": "Nagchowdhury",
  "Id": "0035800000 3YftRAAS"
}
```

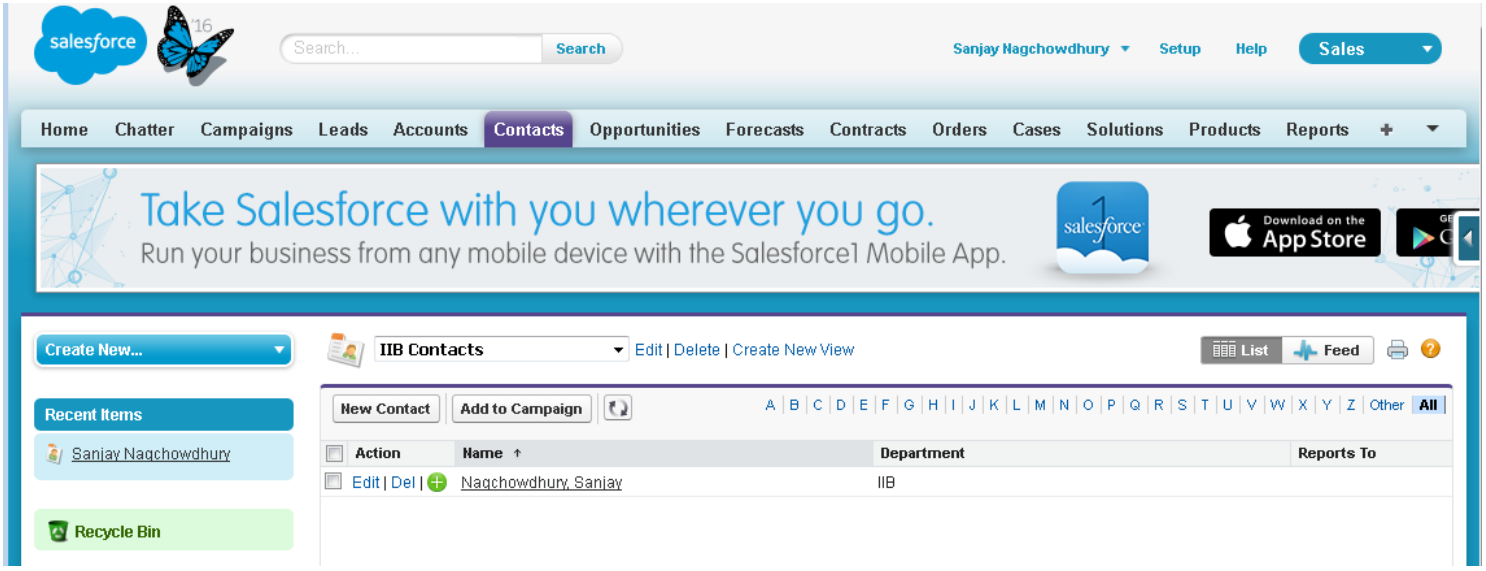
Response Code

```
200
```

Response Headers

```
{
  "content-type": "application/json; charset=ibm-5348"
}
```

If you log onto Salesforce, you can see that the Salesforce contact has been added:



Operations for retrieving a contact, updating a contact or deleting a contact can be implemented in a similar way.

Summary

I have demonstrated new features that have been added in IBM Integration Bus v10.0.0.4. The Salesforce Request node allows you to interact with Salesforce through a message flow. This opens up the possibility of connecting Salesforce with many other endpoints using different transports. The REST API Editor has been enhanced so that you can define a REST API from scratch without needing an existing Swagger document. The Editor will generate a Swagger document for you. The Graphical Data Mapper has been extended so that JSON data types can be modelled using a JSON schema or a Swagger document. I have shown that these three features can be used together if required to access Salesforce through a REST API.

TAGS RESTAPI, IBM-APPLICATION-INTEGRATION, IIBV10.0.0.4, SALESFORCE, AIS-OP

SanjayNagchowdhury

21 comments on"Integrating with Salesforce using a REST API created in IBM Application Integration Suite 1.0"

Syed Imtiyaz Alam October 19, 2019

Hi Thanks for the reply.
So using Salesforce request node we can not do bulk create or bulk update.

Could you please suggest how we can do the bulk create or bulk update as we have a requirement.

Do we need to have SOAP service. Does Salesforce provide any way to do bulk.

Please suggest. Thanks in advance

Reply (Edit)

Syed Imtiyaz Alam October 11, 2019

Hi
i see that in one response , we cannot retrieve multiple records. But
the swagger which is provided by IBM has LIST object also and When i dont pass any Id , it returns all the object(Account)
So is there any way i can retrive more that one Account by passing multiple Ids ?

Reply (Edit)

SanjayNagchowdhury October 14, 2019

Hi,

When using the Salesforce Request node, you can retrieve multiple records by specifying this in the Local Environment in the message flow:
LocalEnvironment.Destination.Salesforce.Request.filter
You can specify a filter with one of the following clauses:
where
limit
skip
order
field

You can control which records are returned by specifying a filter with a where, limit, or skip clause. If no ID has been specified, this filter is used to select the records to be returned.

For more information see these topics in the Knowledge Center:
https://www.ibm.com/support/knowledgecenter/en/SSTTDS_11.0.0/com.ibm.etools.mft.doc/bz90680_.htm
https://www.ibm.com/support/knowledgecenter/SSTTDS_11.0.0/com.ibm.etools.mft.doc/bz90681_.htm

If you are using the flow as a REST application as described in my article, then you would have to tailor the swagger as appropriate.

Thanks

Sanjay

Reply (Edit)

Abe May 24, 2019

Hi Sanjay,
We are trying to map some new fields from SalesForce to dataPower, but the new fields are not reaching datapower.

Can you please let us know what might be the check points.

Reply (Edit)

SanjayNagchowdhury June 07, 2019

Hi Abe,

I'm not absolutely clear on what you are trying to do and where you are finding a problem in your flow. Can you clarify a bit more please.

Thanks

Sanjay

Reply (Edit)

Ansuman January 19, 2018

Thanks, SanjayNagchowdhury
It really helps.

For this task, Is it mandatory to use a paid version of Salesforce account? I am getting an exception like "The REST API is not enabled for this Organization." Could you please guide me on this.

Reply (Edit)

Ian_Larner January 23, 2018

@Ansuman,
I think that you need an API-enabled Salesforce account; that is either the free Developer edition (not the Trial edition) or a paid version with API access.

If you're using a free trial of Salesforce, it won't be API enabled; sign up to the Developer Edition instead.

Regards,
Ian

Reply (Edit)

emarshah September 24, 2017

That's a very good article Sanjay, i am faced with another situation where i have to connect to REST provider who is using oauth1 as authentication method. How can i use security identity to call this REST service using REST Request node, instead of HTTP Request? Creating signature for oauth1 is a real challenge and a lot of code to maintain, if we call it using HTTP Request method. Thanks Sanjay!

Reply (Edit)

Rajat July 17, 2017

Hi Team,

Could you please advise –

I want to transfer account name & Website information from Salesforce to bluemix through api?

Thanks

Reply (Edit)

BenThompsonIBM July 17, 2017

here
Hi Rajat,

Thanks for your comment asking about how to use IBM's integration offerings to expose a Salesforce trigger (eg account name) and invoke

an API, hosted somewhere in Bluemix. We've seen this question has been posed on a couple of articles related to IIB – how to run IIB in a Bluemix container, and also one about how IIB can be used to expose a REST API interface and then call out to Salesforce. IIB is one option in this space, but there is also App Connect (which can expose triggers and actions) which you might want to consider as this provides support for Salesforce events. In fact there is a recent video which we've released that uses Salesforce inputs (triggers) and also happens to showcase how to invoke IIB from App Connect:

There are lots of considerations in this space dependent on your detailed use case, so if you'd like to drop us a private email to discuss specifics then please do so!

Cheers,
Ben

Reply (Edit)

Kiran Ingale May 09, 2017

Hello Sanjay,

Thank you for such a great article! It is very helpful.
We are trying out this node in our environment and I am hitting a limit of 1,000 records when doing a Retrieve operation. I am not setting any where clause in my code, but the limit is set to 200,000.
Our table has over 18 million rows, so I am expecting the Node to return 200,000 records based on the limit I have set in my code.
Is there any configuration property I need to change in IIB or Salesforce to get more number of records when doing the "Retrieve" operation?

Thanks,
Kiran

Reply (Edit)

SanjayNagchowdhury May 10, 2017

When using the the Salesforce Request node to retrieve a large number of records Salesforce imposes a limit on the number of records returned.
If you select with no fields filter from Salesforce, salesforce.com will only send 1000 records back in the request.
If you select with a field filter (so only requesting a subset of fields in the record which are returned) from Salesforce, salesforce.com will send up to 2000 records back in the request.
e.g. in ESQL
SET OutputLocalEnvironment.Destination.Salesforce.Request.filter.field[1].Id = true;
SET OutputLocalEnvironment.Destination.Salesforce.Request.filter.field[2].FirstName = true;
SET OutputLocalEnvironment.Destination.Salesforce.Request.filter.field[3].LastName = true;

Reply (Edit)

Revan January 30, 2017

Thanks for the article Sanjay. It has simplified integration with Salesforce. Can we perform bulk export or import of contacts Salesforce Request Node? Or do we need to use Salesforce Bulk API to perform bulk operations?

Reply (Edit)

SanjayNagchowdhury January 31, 2017

Hi Revan,
You cannot perform a bulk export/import using the Salesforce Request Node. You can use Cast Iron as an alternative which is part of the Application Integration Suite. Otherwise you will need to iterate over your records and issue multiple requests using the Salesforce Request node.

Thanks

Sanjay

Reply (Edit)

BarryT May 09, 2016

Sanjay thanks for the great overview. Was wondering how the salesforce credentials are stored in the broker? Are they encrypted? How would you limit access to the credentials if only certain administrators had permission to supply(or see them)? Thanks,Barry

Reply (Edit)

SanjayNagchowdhury January 31, 2017

Hi Barry,

Yes, the credentials are encrypted. There are stored using mqsisetdbparms. Only Administrators that are a member of mqbrkrs group can run that command.

For more information, see https://www.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/com.ibm.etools.mft.doc/ap08682_.htm

Thanks

Sanjay

Reply (Edit)

Madhsudhana April 21, 2016

Can we do batch updates, UPSERTs and updating data to multiple objects at a time?.

Reply (Edit)

SanjayNagchowdhury January 31, 2017

Hi,

You cannot do batch updates. You can do an upset if you are using an external Id.

Thanks

Sanjay

Reply (Edit)

Asheesh March 25, 2016

we have a competitive situation with the mule soft + Herokuconnect combination . Would love to have your thoughts . Could not find your email id ! Email me at
Asheesh@ sg.ibm.com and we can connect .
Thx

Asheesh

Reply (Edit)

Alex March 23, 2016

Thanks for the great article Sanjay ! Now Salesforce integration is so much easier.

Reply (Edit)

SanjayNagchowdhury March 23, 2016

here
Thanks. Check out the video which demonstrates the integration with Salesforce here:

Reply (Edit)