Configuring Outbound Messages with Harmony API

Introduction

This page describes the recommended approach for configuring outbound messages in Jitterbit – by using a custom Jitterbit Harmony API. This applies to Salesforce outbound messaging as well as to Autodesk evented webhooks and generic webhooks.

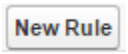
To demonstrate, we use an example of sending an outbound message from Salesforce to Jitterbit that is triggered whenever the Account object is updated in Salesforce. Jitterbit then processes the outbound message request and inserts and/or updates Account information in an Oracle database to keep the information synced. Refer to the Jitterpak Outbound_Message_API_v2.jpk for the example setup.

> **NOTE:** This design pattern uses Design Studio as an example; you may apply the same concepts in Cloud Studio using similar steps.

Step 1 – Create Workflow Rule and Outbound Message to Obtain WSDL (Salesforce)

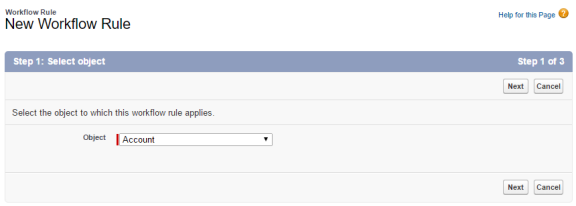
An outbound message must be triggered to call to the Jitterbit Agent (Private or Cloud). In Salesforce, this can be set up using a workflow rule that triggers the outbound message.

1. Log in to your Salesforce account and open your **Workflow Rules**.
   - Go to **Setup** > **Process Automation** > **Workflow Rules**.
   - Or, from **Setup**, use the **Quick Find** box to find **Workflow Rules**.
2. You should be on the screen showing **All Workflow Rules**. Click the button for a **New Rule**.



3. The **New Workflow Rule** screen will take you through a series of steps to set up your new rule. In the first step, **Step 1: Select object**, select the Salesforce object that you will be working with in Jitterbit. Click **Next**.

> **EXAMPLE:** In the example, we used the Salesforce object "Account."



4. Next is **Step 2: Configure Workflow Rule**:
   a. **Edit Rule:** Give your rule a **Rule Name**. It is best practice to include the name of the Salesforce object that the rule applies to in the rule name. Optionally, enter a **Descripti on** of your rule.

   > **EXAMPLE:** In the example, we named our rule "Jitterbit-Account," used to push Salesforce account updates to Jitterbit.

   b. **Evaluation Criteria:** Select the evaluation criteria for your specific use. If you need to create and *not* update, select the first "created" option. If you are going to use synchronization in your operation, select the second "created, and every time it's edited," option. If you have a timed event, select the third option ,"created, and any time it's edited to subsequently meet criteria."

   > **EXAMPLE:** In the example, we selected "created, and every time it's edited" because we want the Salesforce "Account" object to synchronize.

   c. **Rule Criteria:** In order to prevent an infinite loop when objects are updated, it is strongly suggested to exclude the Salesforce username that is configured on the Salesforce Org in Jitterbit. Under **Field**, use the dropdown to select "Current User: Username," with the **Operator** specified as **"**not equal to," and the **Value** as the Salesforce username for the Salesforce Org used in Jitterbit. Click **Save & Next** to continue.

On This Page

- Introduction
- Step 1 – Create Workflow Rule and Outbound Message to Obtain WSDL (Salesforce)
- Step 2 – Configure Operations (Design Studio)
  - Create Script and Global Variable
  - Create API Response Operation
  - Create Upsert Operation
- Step 3 – Create Custom API (Jitterbit API Manager)
- Step 4 – Update the Endpoint URL (Salesforce)
- Step 5 – Activate Outbound Message Trigger (Salesforce)
- Other Ways to Configure Outbound Messages

Jitterpak

Outbound_Message_API_v2.jpk

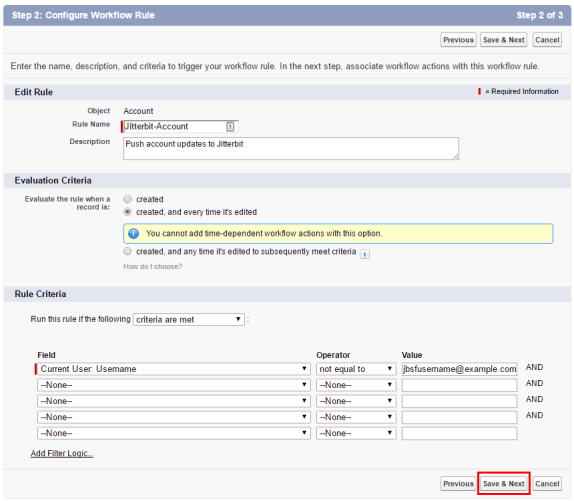
Related Articles

- API Jitterbit Variables
- Capturing Data Changes with a Harmony API or HTTP Endpoint
- Configuring Outbound Messages with Hosted HTTP Endpoints

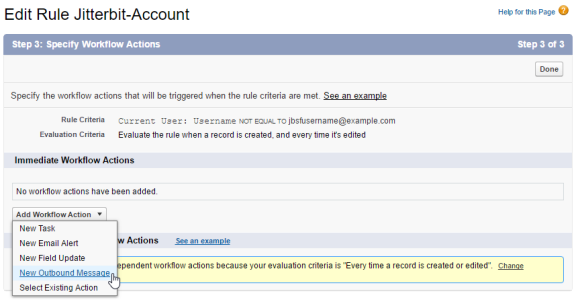
Related Topics

- API Manager
- Design Pattern Library
- Design Studio
- Salesforce Wizard Overview
- Sources and Targets

Last updated: Apr 19, 2019



5. Next, in **Step 3: Specify Workflow Actions**, you will need to define the action that should happen when the rule is triggered. Under **Immediate Workflow Actions**, use the **Add Workflow Action** dropdown to select **New Outbound Message**.



   a. Give your outbound message a **Name** and optionally a **Description**.

   > **EXAMPLE:** In the example, we named our outbound message "Jitterbit-Account Update," defining the fields to send to Jitterbit.

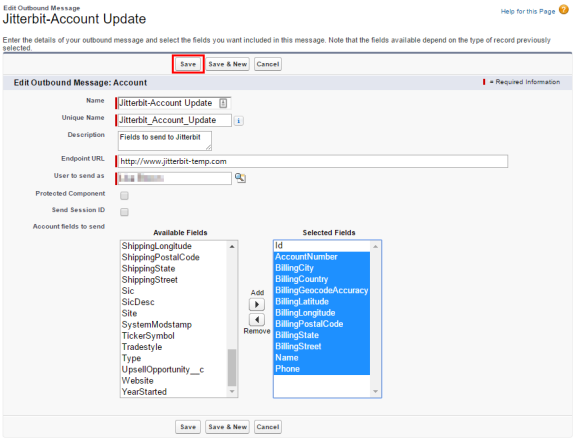
   > **NOTE:** The **Unique Name** can be left as the name that is automatically generated unless another outbound message with the same unique name has previously been used on your Salesforce account.

   b. For the **Endpoint URL**, enter a dummy URL that will be replaced later on in this process. In a later step described on this page, you will need to replace this temporary URL with the final API URL that will be provided by Jitterbit during creation of your custom Harmony API.

   > **EXAMPLE:** In the example, for now we use a dummy Endpoint URL of `http://www.jitterbit-temp.com`.

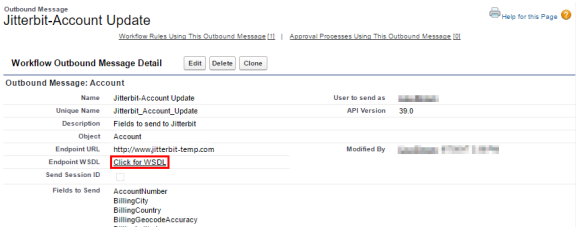
   c. Select the **Available Fields** that you want to use in your Jitterbit operation and add them to the **Selected Fields** side using the arrows. The "Id" field will be selected automatically. When you are done, click the **Save** button.

   > **NOTE:** It is recommended to include all desired fields that you want to sync in the outbound message in order to cut down on the number of API requests.
   >
   > This is because a workflow rule can apply to only one object in Salesforce. Therefore an outbound message is for a single object. Note that you cannot use an outbound message to edit and then send the account and all the contacts into a database. Instead you can send the ID field in the outbound message into Jitterbit, and then query back into Salesforce to pull all the desired records (see Salesforce Query Wizard).



6. This should return you to the **Step 3: Specify Workflow Actions** screen. From here, within **Immediate Workflow Actions**, click on the name of the outbound message you have just created. On the next screen, under **Workflow Outbound Message Details**, next to **Endpoint URL**, use the "Click for WSDL" link and save the file, being sure to change the file extension from .xml to .wsdl. You may also wish to change the name of the WSDL file to reflect the specific outbound message.

> **EXAMPLE:** In the example, we named our WSDL file "`jitterbit-account-outbound.wsdl`".



> **NOTE:** Even though the **Endpoint URL** is temporary, to be replaced with the final URL during a later step, it's OK to download the WSDL now because the URL is not used by Jitterbit. However, if you prefer, you can come back and re-download the WSDL later so that the final URL is listed in the file.

Step 2 – Configure Operations (Design Studio)

Next you need to create an operation chain in Jitterbit that will handle the Harmony API call. This can be done a variety of ways. Below is a recommended setup for a chain of operations that takes into account best practices in Jitterbit.

Create Script and Global Variable

In this demonstration, the first operation in the operation chain will be a script. This script operation will be used to receive the Salesforce outbound message.

> **NOTE:** Using a script as the first operation is not required, but is recommended because the first operation in a chain is not logged unless debugging is turned on.
>
> Therefore, we want to put our API processing downstream in the operation chain so that if there are any issues, we can more easily troubleshoot, as well as see a more complete history of errors, without turning on debugging.

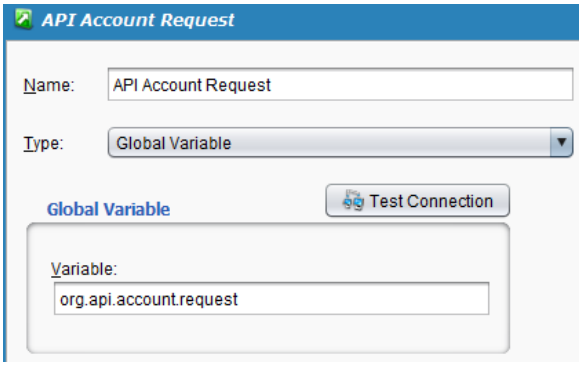
1. Within your Design Studio project, create a new script as an operation. This can be done in several ways described in Creating an Operation. One way is to right-click on **Operations** in the project items list on the left, and and select **New Operation**. In the popup, select **Script** as the type.
2. The **Operations** tab will open in the main view of Studio with a graphical representation of your operation containing a **Script**. Double-click the **Script** icon within the operation and select the button **Create New Jitterbit Script**.

3. The **Scripts** tab will open. In the script, we will use a new global variable for the API request. Between the `<trans>` tags, enter the following. Also give your script a **Name** (e.g. "Receive SF Account Outbound Message"), then save and close the **Scripts** tab.

```
<trans>
$org.api.account.request=$jitterbit.api.request.body;
</trans>
```

> **EXAMPLE:** In the above example, we chose to name our new global variable `org.api.account.request` and set it equal to the predefined global variable `jitterbit.api.request.body`. For more information see Global Variables and API Jitterbit Variables.

4. Since we are using a new global variable in the script, we also need to create the global variable in Jitterbit. This can be done in several ways described in Global Variable Source. One way is to right-click on **Sources** in the project items list on the left, and select **New Source**. In the popup, select **Global Variable** as the type.
5. The **Sources** tab will open in the main view of Studio with the **Global Variable** type selected. Enter a **Name** for your global variable (e.g. "API Account Request") and enter the **Variable** you want to create as used in your script.



6. The first operation in the operation chain is now configured.

Create API Response Operation

The next operation in our operation chain will be used to acknowledge the API response. This operation will be set up to run on a successful run of the first operation (when a Salesforce outbound message is received).

> **NOTE:** This operation is used to send a success message back to Salesforce immediately, indicating that the outbound message has been received. This response does not indicate if the database update was successful.
>
> If you need to send success or failure messages to Salesforce, you can alternatively put the API response at the end of the operation chain and provide a global variable. However, note that for long-running operations this is more susceptible to timeout issues. Instead, we suggest sending an acknowledgement as soon as the request is received, as described here.

1. Within your Design Studio project, create a new transformation as an operation. This can be done in several ways described in Creating an Operation. One way is to right-click on **Operations** in the project items list on the left, and and select **New Operation**. In the popup, select **Transformation** as the type.
2. The **Operations** tab will open in the main view of Studio with a graphical representation of your operation containing the standard components: **Source**, **Transformation**, and **Target**. Double-click the **Transformation** icon within the operation and select the button **Create New Transformation**.
3. The **Transformation Wizards** tab will open where you will configure the operation.
    a. On the first screen, **Name**, enter a **Name** for the transformation (e.g. "Send API Response"), select a **Source** of the outbound message if applicable, and select the **Target** of the API response. Click **Next** to continue.

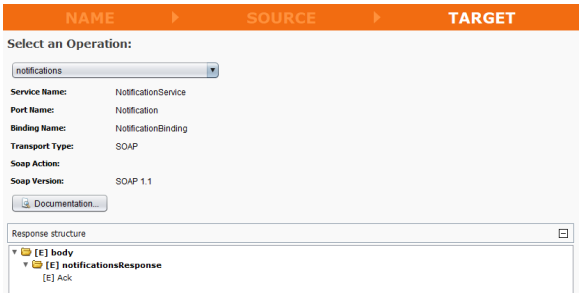
> **EXAMPLE:** In the example, we do not need to configure a source because the API request from the Salesforce outbound message is being received by our script operation. In this case the selected **Source** is "(None)." For **Target** , we selected "SOAP/WSDL Response" to use the WSDL we downloaded from Salesforce.

b. On the next screen, **Target**, select the button for "Select a local WSDL file" and browse to the location where you saved the WSDL downloaded from Salesforce.

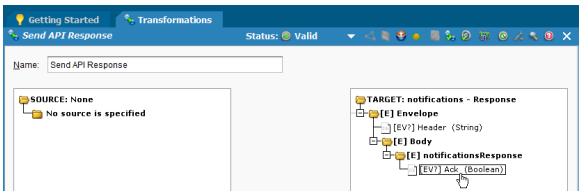
> **NOTE:** The **Source** screen of the **Transformation Wizard** is no longer applicable when the **Source** is set to "(None)."

c. On the same **Target** step, you can now select an operation if multiple methods are available. You can determine which one you need by looking at the request and/or response structures. Click **Next** to continue. Then choose whether to validate against the WSDL's schema, and click **Finish** to continue.

> **EXAMPLE:** For a Salesforce outbound message, nothing needs to be modified, so we just click **Next**.



4. The **Transformations** tab will display. On the right side (target side), fully expand the tree and double-click on "[EV?] Ack (Boolean)."

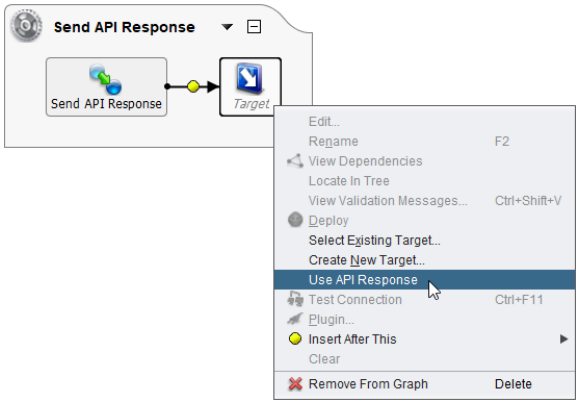


5. The **Formula Builder** will display. Type the word "true" (no quotes) between the `<trans>` tags and click **OK**. Then save and close the **Transformations** tab.

```
<trans>
true
</trans>
```

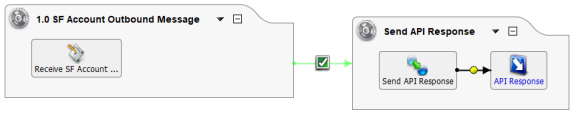
> **NOTE:** The "true" lets Salesforce know the record was received successfully. If "true" were not received, then Salesforce would continue trying to send the record for 24 hours increasing the time between attempts.

6. Now return to the **Operations** tab and you should see your configured transformation, along with a **Target**. Right-click on the **Target** icon and select **Use API Response**. If you have not done so already, double-click on the operation name to rename it (e.g. "Send API Response").



EXAMPLE: In the example, the **Source** icon is now removed from the operation graph since we selected a source of "(None)" in the transformation.

7. Next, we need to set up this operation to run when our first configured operation (with the script) is successful. To do so, return to the **Operations** tab and select your first operation. Right-click on the operation or on the down arrow next to the operation title, and select **On Success** > **Operation** > **Select Existing**. In the popup, select the operation you just created to send the API response (e.g. "Send API Response"). Then click **OK**.
8. These operations should now run as an operation chain. When the first operation is successfully run (when a Salesforce outbound message is received), an acknowledgement will be sent back to Salesforce.



Create Upsert Operation

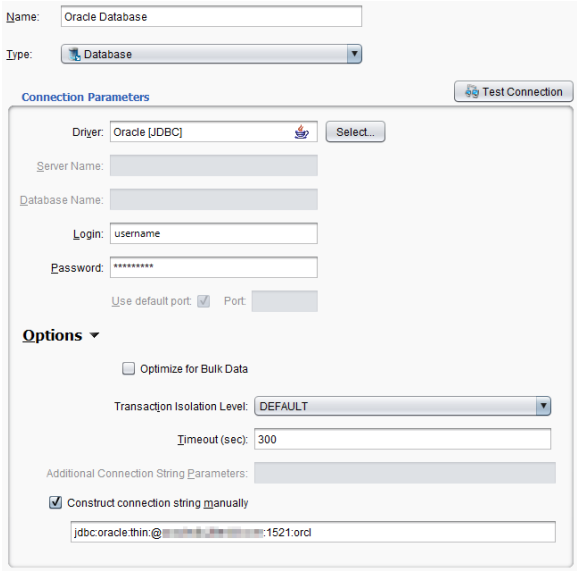
Next you are ready to configure another operation that will write to the database. This operation will be set up to run dependent upon the first operation running (when a Salesforce outbound message is received).

1. Within your Design Studio project, create another new transformation as an operation. This can be done in several ways described in Creating an Operation. One way is to right-click on **Operations** in the project items list on the left, and and select **New Operation**. In the popup, select **Transformation** as the type.
2. The **Operations** tab will open in the main view of Studio with a graphical representation of your operation. Double-click on the operation name to rename it (e.g. "Upsert Oracle Account"). Within the new operation, double-click the **Source** icon and select the existing global variable source that was created for use in the script from the first operation. This is the API request that will be used for the upsert.

EXAMPLE: In the example, we selected our global variable, named "API Account Request."

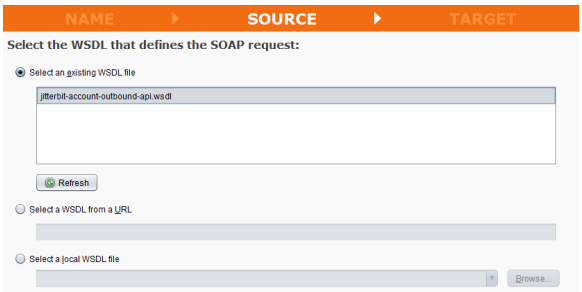
3. Within the new operation, double-click the **Target** icon and configure for your desired final target. Select an existing target that is configured for the Salesforce data that you are pulling, or create a new target. See additional instructions for targets on Sources and Targets.

EXAMPLE: In the example, we use an Oracle database as the target.

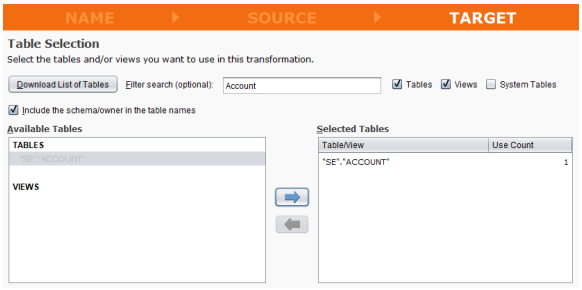


4. Within the new operation, double-click the **Transformation** icon in the operation and select the button **Create New Transformation**. The **Transformation Wizard** will open.
    a. On the **Name** screen, give your transformation a **Name** (e.g. "Upsert Oracle Account"). For **Source**, choose **SOAP/WSDL Request**. Click **Next** to continue.

    b. On the **Source** screen, select the same WSDL that you used earlier in this process. It should be listed under "Select an existing WSDL file." Click **Next** to continue.



    c. Unless modifications are necessary, click **Next** and **Next** again to advance to the **Target** screen. Since we are using a database as the target, click the button to **Download List of Tables**. Filter or look through the **Available Tables** and select the one you want to use, then use the arrows to add it to the **Selected Tables** list. Click **Next** to continue.

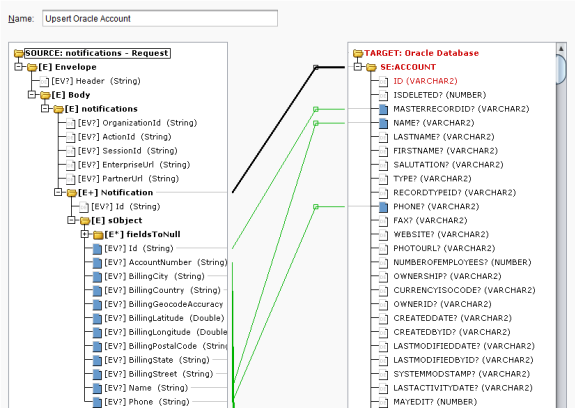


    d. Next, choose the **Insert/Update Mode** that fits your needs. **Insert/Update** is for new records and record updates, **Insert Only** is for new records only, and **Update Only** is for record updates only. Then click **Finish** at the bottom of the screen.

> **EXAMPLE:** In the example, we chose **Insert/Update** (upsert) because we want to create new records and update existing records.

5. The **Transformations** tab will open, allowing you to create your mappings by dragging and dropping from the source (left side) to the target (right side). After you create the mappings, be sure to save.

> **NOTE:** The red text in the image below indicates a required field; however, it's OK that this field is not mapped because the ID field will be generated on the database side.

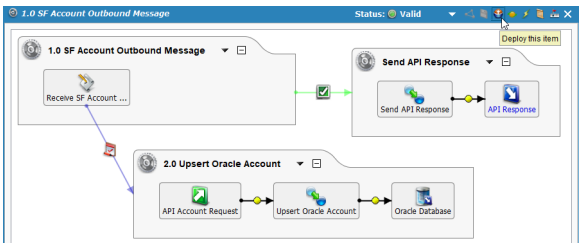


6. Next, we need to set up this operation to run when our first configured operation (with the script) receives a Salesforce outbound message. To do so, return to the **Operations** tab and select your first operation. Double-click on the script to re-open it. Add a `RunOperation` function at the end of your existing script, so that the final script looks similar to the following, replacing the name of your global variable and name of the operation you want to run.

```
<trans>
$org.api.account.request=$jitterbit.api.request.body;
RunOperation("<TAG>Operations/2.0 Upsert Oracle Account</TAG>")
</trans>
```

7. The first and second operations should now run as an operation chain. When a Salesforce outbound message is received, an acknowledgement will be sent back to Salesforce. You are now ready to deploy the project to your selected Harmony Environment. Click the deploy icon.

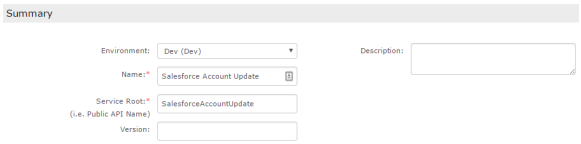
> **NOTE:** Don't stop here. There are several steps to complete after deploying!



Step 3 – Create Custom API (Jitterbit API Manager)

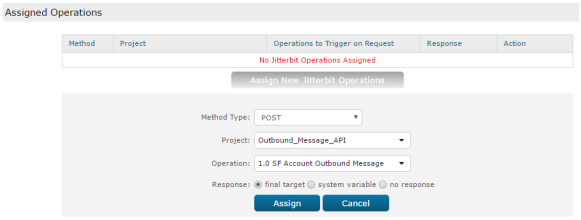
Now that the project is deployed, you can create your custom Jitterbit Harmony API that will be exposed to receive the outbound messaging data from Salesforce.

1. Log in to the Harmony Portal and go to **API Management** > **Custom API**.
2. Click the button **New Custom API Service**. On the resulting page, the following fields were configured for the example used in this demonstration. Refer to API Manager for additional documentation on all available options.
    a. Under **Summary**, the following fields were configured.



    - **Environment:** Use the dropdown to select the environment where your Jitterbit Studio project containing the operations above was deployed.
    - **Name:** Enter a name for your custom API.
    - **Service Root:** Enter a name that will be included in the exposed API URL in the format of https://BaseURL/Environment/ServiceRoot. This field will be prepopulated based on the API name you provide, although you can change it here if desired.

    b. Under **Assigned Operations**, the following fields were configured. When all fields are filled out, click the **Assign** button to add the operation to the list.



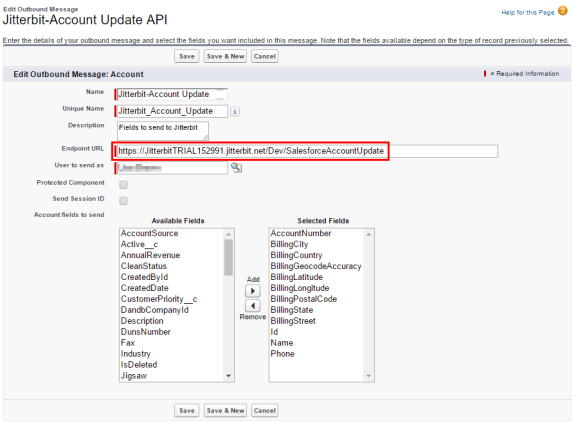
    - **Method Type:** Use the dropdown to select the type of method.
    - **Project:** Use the dropdown to select the name of the Jitterbit Studio project containing the operations you deployed above.
    - **Operation:** Use the dropdown to select the desired operation within your Jitterbit Studio project.
    - **Response:** Select the type of response.

    c. All other settings are kept as default for this example. Click **Add** at the bottom of the page to create the new custom API.
3. After your custom is created, you will now see it listed in a table on the **API Management** > **Custom API** page. Under the **Description** column, the custom API URL is provided. This is what you will need to use to update the **Endpoint URL** in Salesforce.

Step 4 – Update the Endpoint URL (Salesforce)

Now that you have the URL from your custom API, you need to update the **Endpoint URL** on your outbound message in Salesforce to use it.

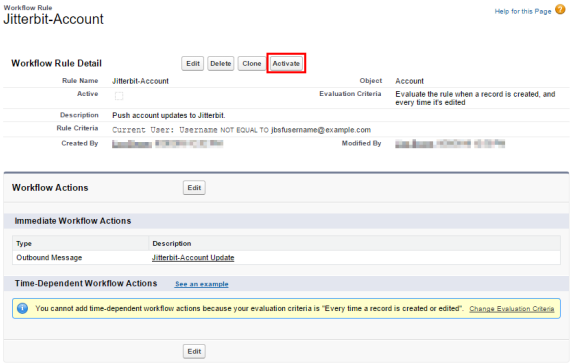
1. Log in to your Salesforce account and open your **Outbound Messages**.
    - Go to **Setup** > **Process Automation** > **Workflow Actions** > **Outbound Messages**.
    - Or, from **Setup**, use the **Quick Find** box to find **Outbound Messages**.
2. You should be on the screen showing **All Outbound Messages**. Click on the name of the outbound message you previously set up to view its details.
3. Under **Workflow Outbound Message Detail**, click the **Edit** button. You can now replace the dummy URL entered in the Endpoint URL field with the actual URL of the custom API you just created. Click **Save** to update.



Step 5 – Activate Outbound Message Trigger (Salesforce)

You are now ready to turn on the rule that triggers the outbound message to Jitterbit.

In Salesforce, return to your **Workflow Rules** and click on the rule you would like to use. Then click the **A ctivate** button to activate the trigger.



Other Ways to Configure Outbound Messages

The approach described on this page using a custom Jitterbit Harmony API is the recommended approach to configuring outbound messages. An additional example using a custom Jitterbit API is provided at Capturing Data Changes with a Harmony API or HTTP Endpoint.

An additional approach to configure outbound messages uses Hosted HTTP Endpoints. Refer to Configuring Outbound Messages with Hosted HTTP Endpoints for additional information.