

# Vonage Conversations for Salesforce

## Getting Started Guide

[Pre-Requisites](#)

[Deployment](#)

[First Time Deployment](#)

[UI Setup](#)

[Add the widget to a standard sObject](#)

[Add the widget to a custom sObject](#)

[Sending Messages from the UI](#)

[Sending an SMS or WhatsApp Message](#)

[Sending a Message with a Resource](#)

[Flow Development](#)

[Send Automatic Response](#)

[The platform event trigger for initiating the Flow:](#)

[Is it an incoming message?](#)

[Get the message details:](#)

[Is the message incoming via WhatsApp?](#)

[Is the message incoming via SMS/MMS?](#)

[Send back a WhatsApp message](#)

[Is it a test request?](#)

[Send text response](#)

[Send image response](#)

[Troubleshooting and Support](#)

[Vonage Useful links](#)

[Analyze](#)

[Support Information](#)

## Pre-Requisites

This guide will explain how to deploy the Vonage Conversations for Salesforce package and send an SMS/MMS message using the UI.

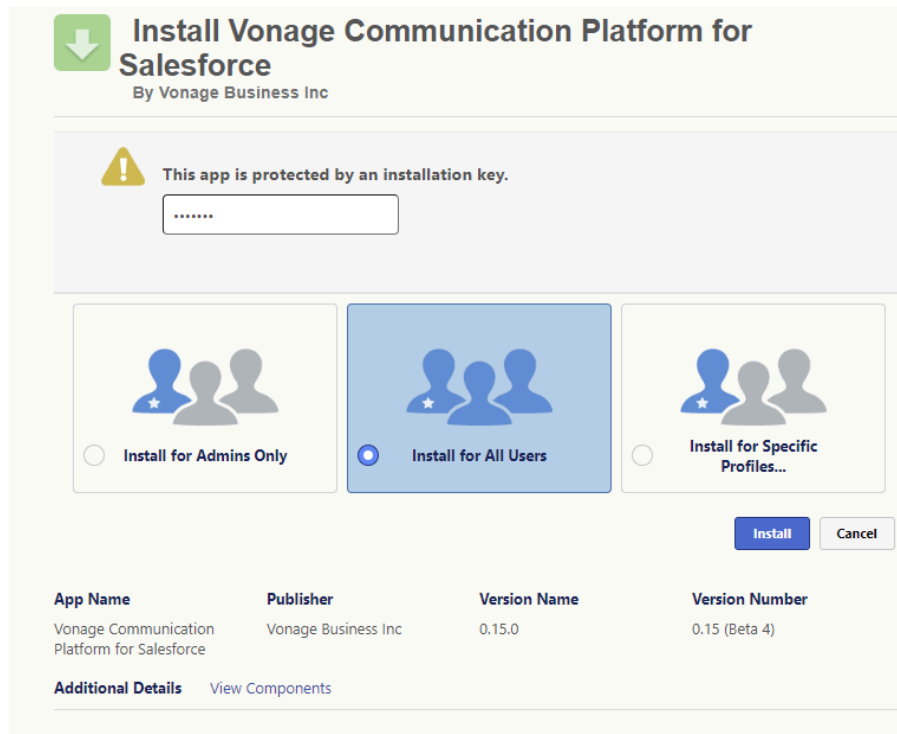
1. A [Vonage account](#) with your **api-key**.

2. To use WhatsApp, your external account should be configured and approved for use with Vonage; if you are a managed customer, [you can follow this guide](#), or if you are non-managed [you can follow our self-service guide](#).
3. Note: MMS messaging is only available in the US & Canada. To use images, your LVN must be a USA/CA number that is approved for [10 DLC](#).
4. For RCS onboarding support, please connect with your Vonage account manager or point of contact.


## Deployment

To install the package, you will need the deployment URL and installation key. These will be provided by your Vonage point of contact.

First, go to the deployment URL and login to your Salesforce organization. You will see this screen:



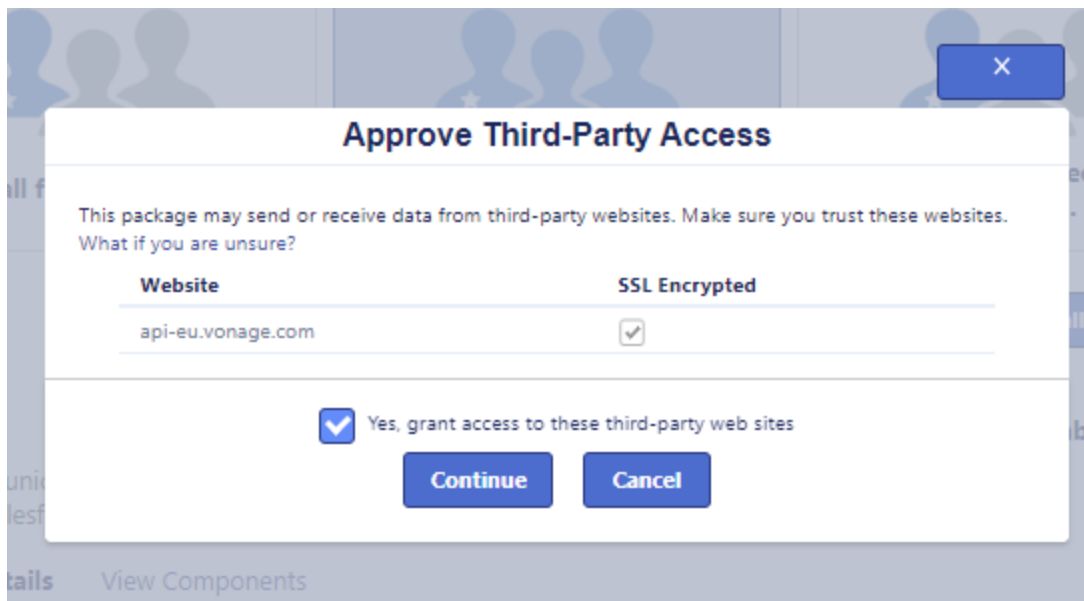
Select the group that you want to install the package for, and enter the installation key provided. The warning below may appear. If it does, select "Done" and wait for the package deployment to complete.



**This app is taking a long time to install.**  
You will receive an email after the installation has completed.

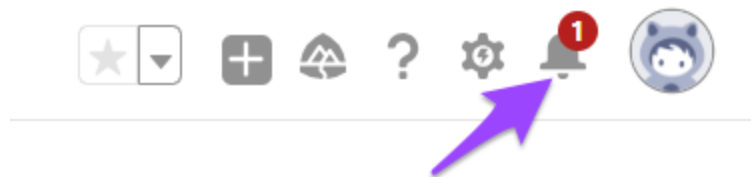
App Name	Publisher	Version Name	Version Number
Vonage Communication Platform for Salesforce	Vonage Business Inc	0.15.0	0.15 (Beta 4)

Should the following window popup, please grant access, as the URL is used by the package to access Vonage services:

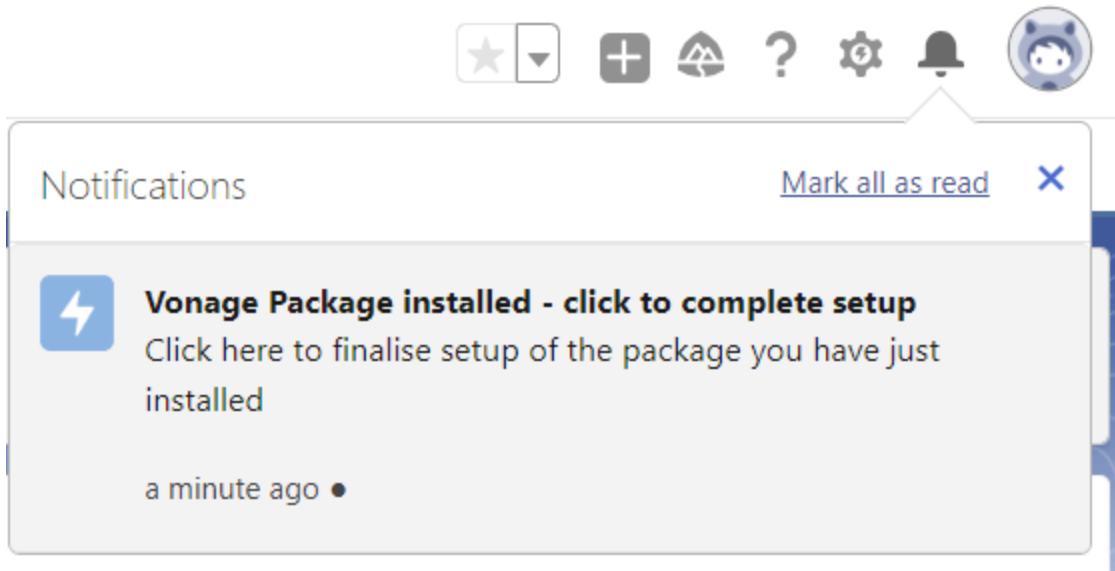


## First Time Deployment

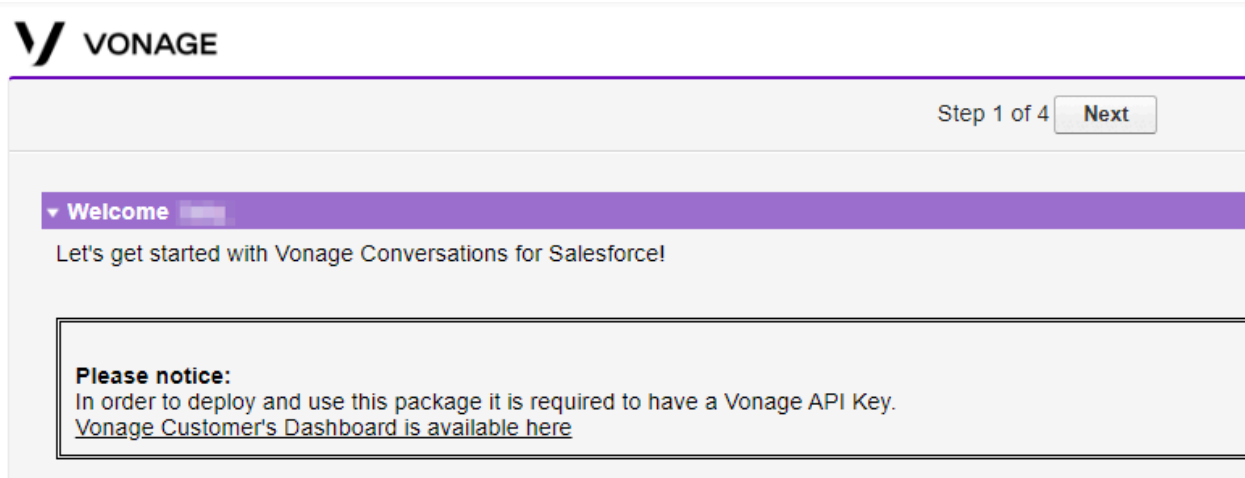
Once the initial deployment is done, the 'Bell Notification' will indicate a new message:



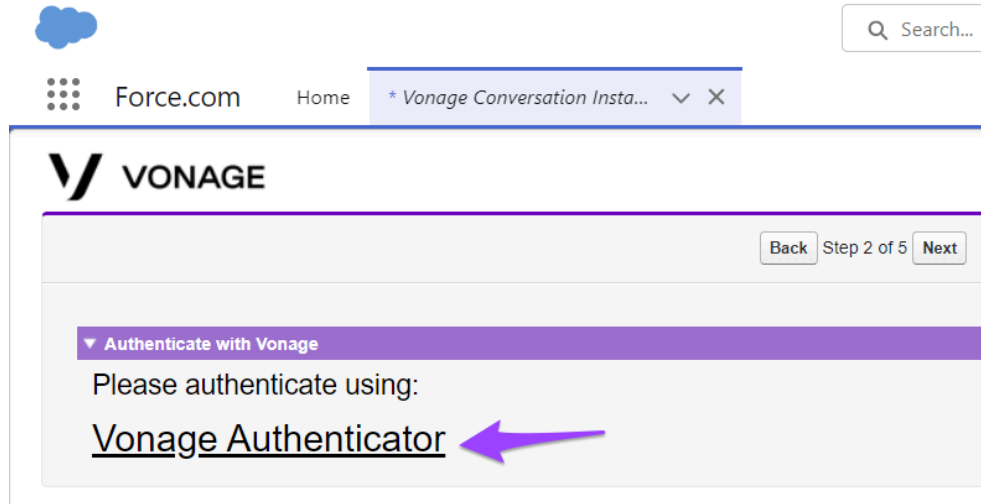
Select the link in the message to complete the setup:



Click 'Next' to start the deployment process:

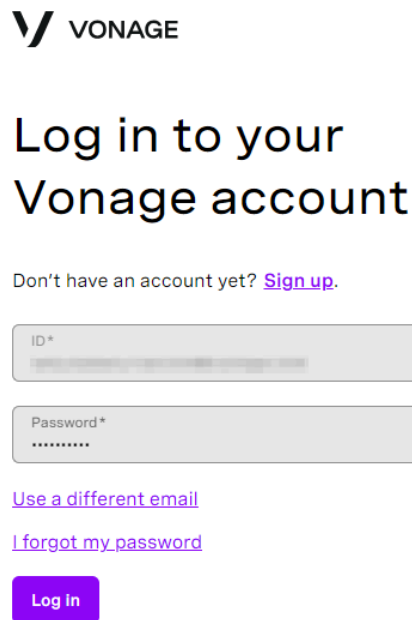


For Step 2, click the link to authenticate with Vonage - please ensure that you are logged-off from any other Salesforce ORG while completing this process:

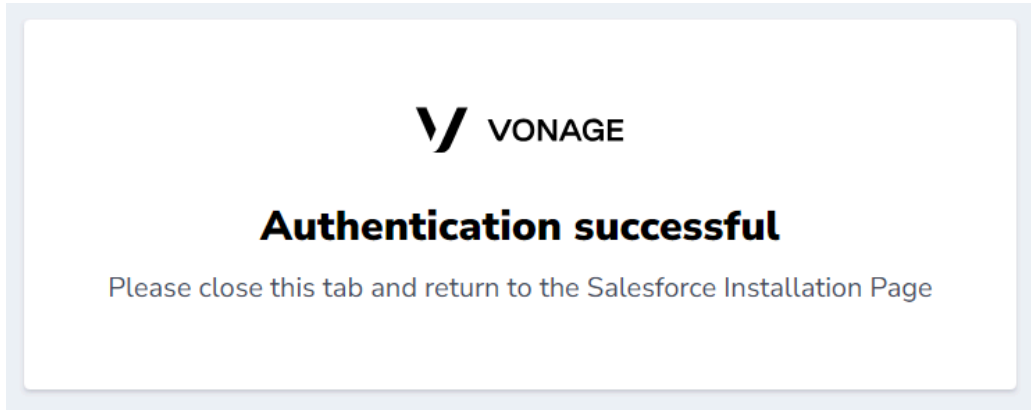


**Failure to complete the authentication will cause errors in later stages of the wizard.**

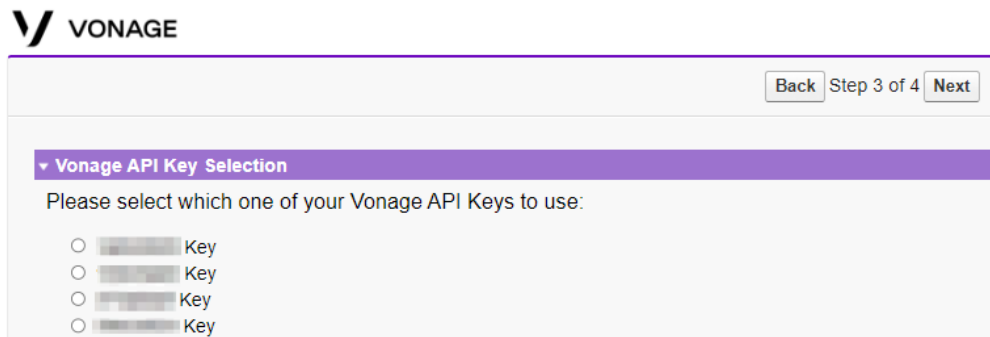
Next, you will be taken to the Vonage customer dashboard to login:



Once authentication is completed, you will see this confirmation and can return to Salesforce:



Step 3 of the installation wizard contains a link back to the Vonage customer dashboard, where you can configure your account, add API keys, and rent additional LVNs:



In step 4, select the API key you want to use:

▼ Vonage API Key Selection

Please select which one of your Vonage API Keys to use:

- [Redacted] Key
- [Redacted] Key
- [Redacted] Key
- [Redacted] Key
- [Redacted] Key
- [Redacted] Key
- [Redacted] Key
- [Redacted] Key
- [Redacted] Key

The final step will allow you to select one of the following options:

- The default option is to continue without selecting any endpoint. Later, using the Admin application you can add your Brand or select any of your available numbers and external accounts.
- Alternatively, select one of the numbers or external accounts associated with the selected Vonage API Key.

▼ **Number Selection**

Please select one of Vonage APIKey [redacted] End-Points to use, then click the **Finish** button to complete the setup.

**Please be aware: Endpoints cannot be shared across different Salesforce ORGs.**

This is a long process. Please do not close or refresh this window before the operation is completed.

- Use alphanumeric sender name or add a brand endpoint later.
- 12[redacted] - Country: US, supported channel: MMS
- 44[redacted] - Country: GB, supported channel: SMS
- 44[redacted] - Country: GB, supported channel: SMS
- [redacted], supported channel: RCS, external name: Vonage RCS Demo - [redacted] external account id: [redacted]
- 44[redacted], supported channel: WhatsApp, external name: [redacted], external account id: [redacted]
- 44[redacted], supported channel: WhatsApp, external name: [redacted], external account id: [redacted]

After the initial deployment it will be possible to add additional numbers.

Once this is complete, setup is done and you will see the following message:

▼ Post Setup

The setup is done!

Please assign the **Vonage Connector User Permissions Set** to the relevant users.  
The **Vonage Conversation** Lightning Web Component is ready to be added to any record page.

### Release Notes

**\*\* Release 0.14 \*\***

Target: Early access customers

Minor fixes:

1. WhatsApp templates configuration save failure
2. Rolling log cleanup

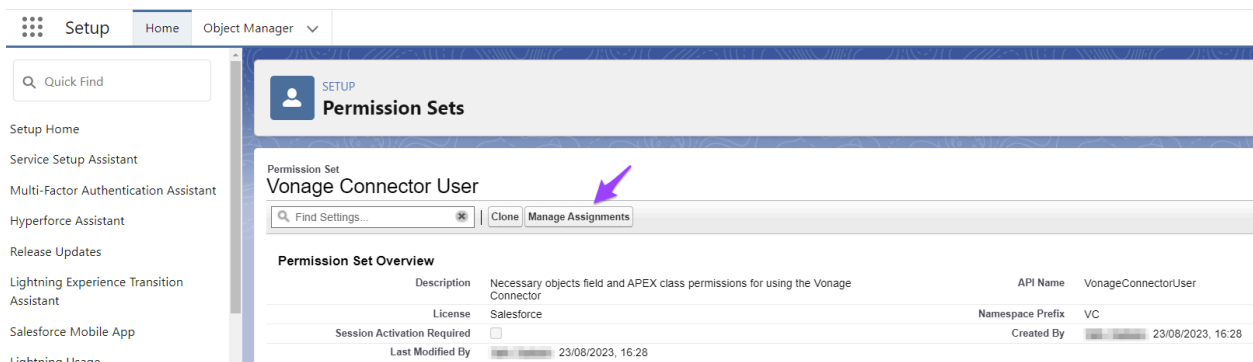
**\*\* Release 0.13 \*\***

Target: Early access customers

New Features:

1. WhatsApp templates support
2. External channel support
3. Delivery status support

The next step is to set the permissions for which users can use the Vonage package. To do this, navigate to 'Permission Sets', then select 'Vonage Connector User' and click Manage Assignments:



Setup Home

Service Setup Assistant

Multi-Factor Authentication Assistant

Hyperforce Assistant

Release Updates

Lightning Experience Transition Assistant

Salesforce Mobile App

Lightning Usage

Permission Sets

Permission Set: Vonage Connector User

Find Settings... Clone Manage Assignments

**Permission Set Overview**

Description	Necessary objects field and APEX class permissions for using the Vonage Connector	API Name	VonageConnectorUser
License	Salesforce	Namespace Prefix	VC
Session Activation Required	<input type="checkbox"/>	Created By	23/08/2023, 16:28
Last Modified By			23/08/2023, 16:28

Assign the permission set to the agents you want to use the Vonage package to complete deployment.

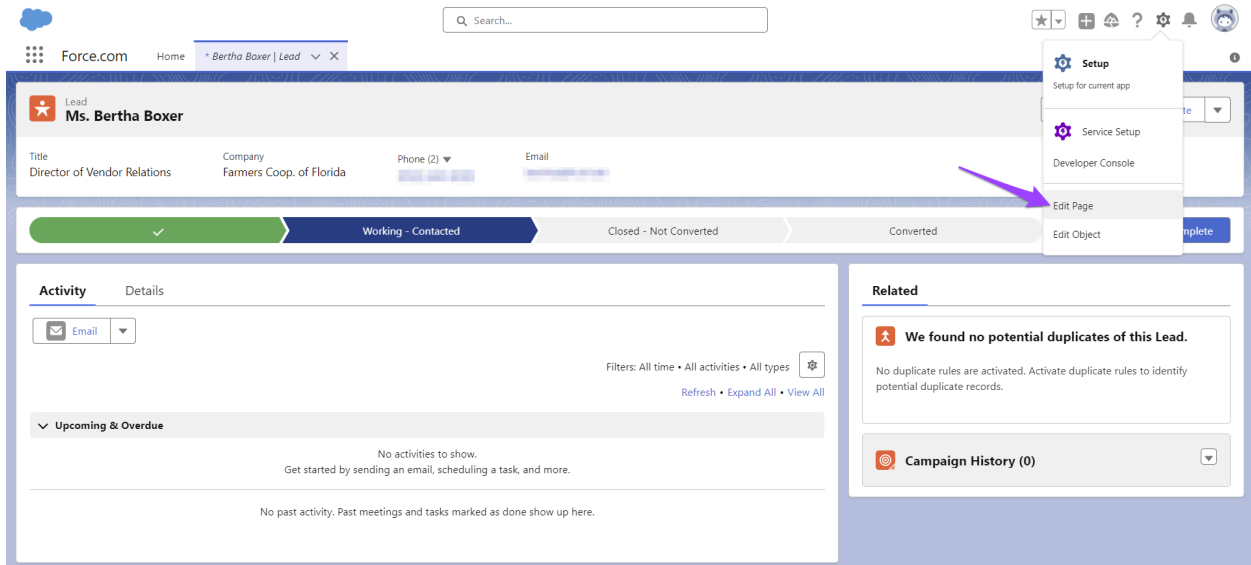
The next section will explain how to add the UI widget to a record page.

## UI Setup

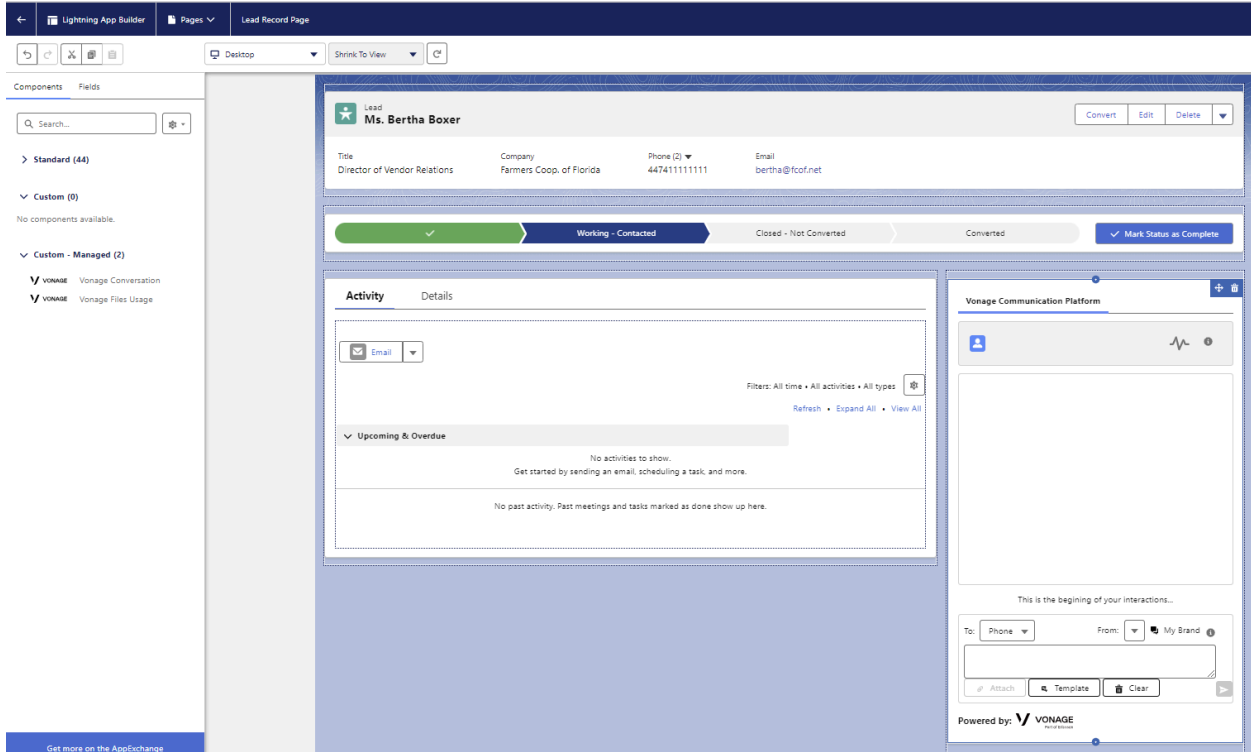
The UI Widget can be added to the record page of any standard sObject or custom sObject.

### Add the widget to a standard sObject

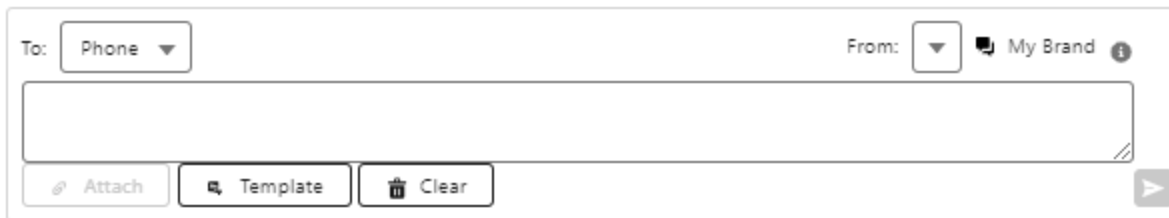
To add the widget to a standard sObject, go to the record page and select "edit page":



Drag the widget to the suitable place on the page using the components menu on the left. For example, editing the page for a lead would take you to this screen:



The widget will automatically identify all the populated "Phone" type fields in the record and will add them to the drop down to allow the agent to select which number to send messages to:



Note that the drop down uses the field label and not the phone number itself in order to allow the admin to hide the original phone numbers if they wish to do so.

Finally, save to activate the widget on the record page. The agents with the Vonage permissions set assigned will be able to see the widget and start sending messages:

Vonage Communication Platform

David Monaco

07-Oct-2024

David Monaco  
Hello, I would like to book a car service. Reg: AA12XYZ

Thank you, will 21st July 8am be suitable for you?

To: Phone From: 44

Attach Template Clear

Powered by: VONAGE

Vonage Communication Platform

Marc Smith

17-Oct-2024

Marc Smith  
Thank you!

Tally Tabary  
Hi Marc  
Your order: 1554424 has been successfully delivered  
Any questions, we are here to help!

To: Phone From: 44

Attach Template Clear

Powered by: VONAGE

Vonage Communication Platform

Bertha Boxer

07-Oct-2024

Bertha Boxer  
Hello, I would like to set an appointment

Sure - how about 12th June 3pm?

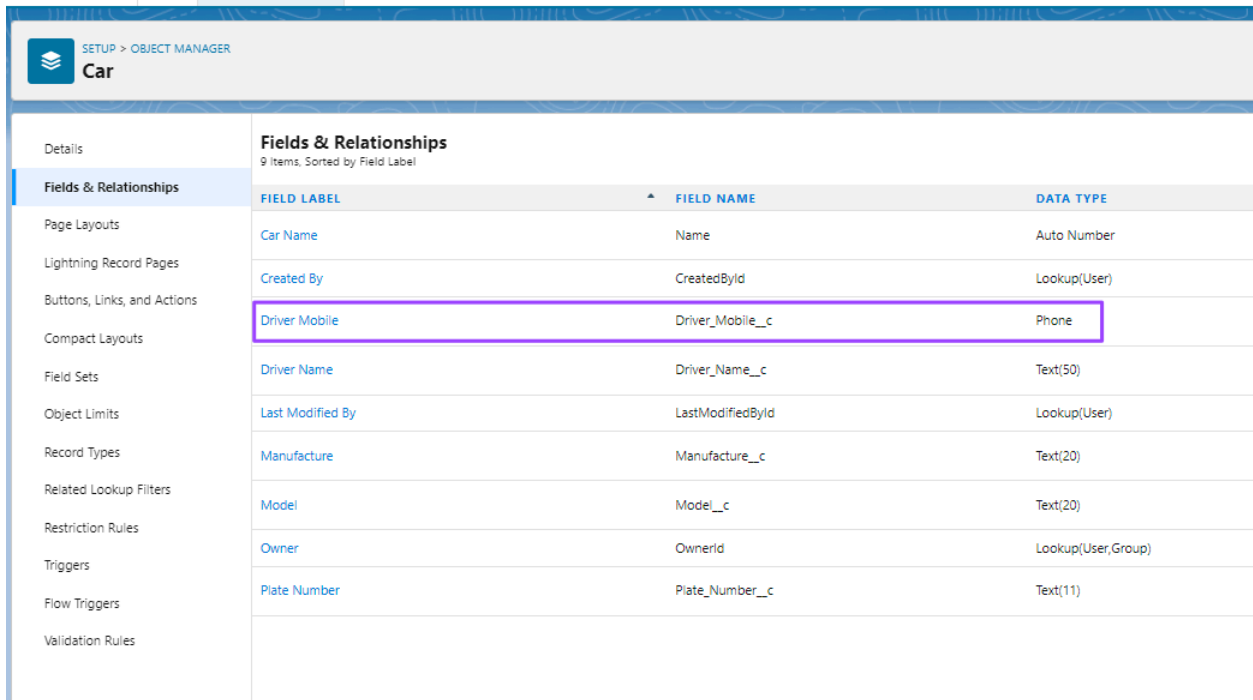
To: Phone From: 44

Attach Template Clear

Powered by: VONAGE

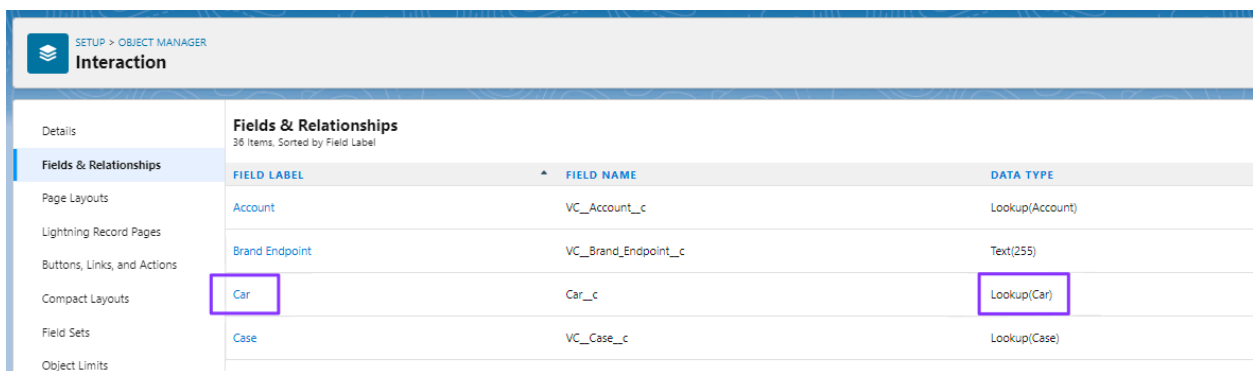
## Add the widget to a custom sObject

In this example, we'll add the UI to a custom sObject named **Car\_\_c**. The sObject must include at least one field of type "Phone", such as 'Driver Mobile' here:



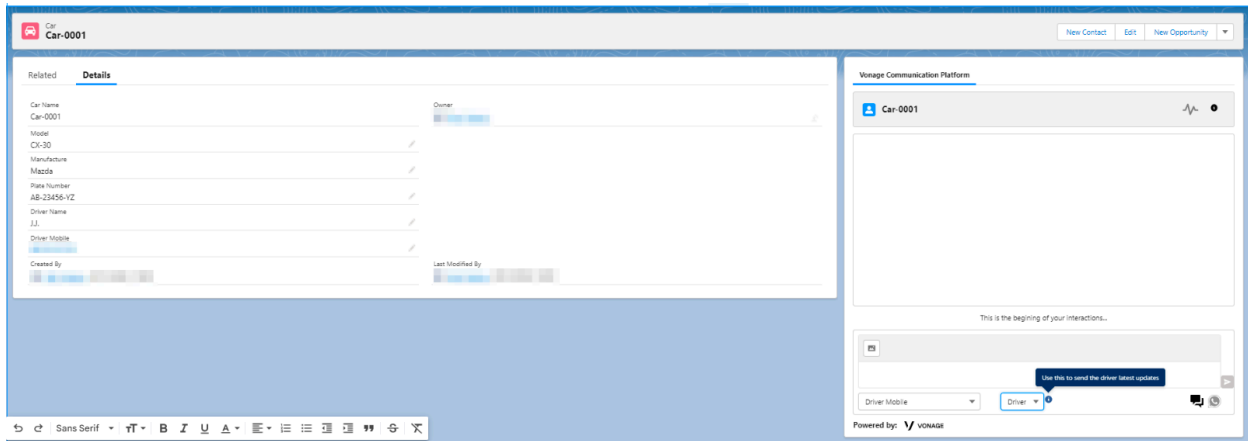
FIELD LABEL	FIELD NAME	DATA TYPE
Car Name	Name	Auto Number
Created By	CreatedById	Lookup(User)
Driver Mobile	Driver_Mobile__c	Phone
Driver Name	Driver_Name__c	Text(50)
Last Modified By	LastModifiedById	Lookup(User)
Manufacture	Manufacture__c	Text(20)
Model	Model__c	Text(20)
Owner	OwnerId	Lookup(User,Group)
Plate Number	Plate_Number__c	Text(11)

The **VC\_Interaction\_\_c** sObject should also contain a field that has a lookup relationship with the **Car\_\_c** sObject:



FIELD LABEL	FIELD NAME	DATA TYPE
Account	VC_Account__c	Lookup(Account)
Brand Endpoint	VC_Brand_Endpoint__c	Text(255)
Car	Car__c	Lookup(Car)
Case	VC_Case__c	Lookup(Case)

Once that's in place, it is possible to add the widget to the **Car\_\_c** record page:



The screenshot shows a Salesforce record page for 'Car-0001'. The left sidebar contains a 'Details' section with the following fields:

- Car Name: Car-0001
- Model: CX-30
- Manufacture: Mazda
- Plate Number: AB-23456-VZ
- Driver Name: J.J.
- Driver Mobile: [Redacted]
- Created By: [Redacted]
- Last Modified By: [Redacted]

The main content area on the right features a 'Vonage Communication Platform' widget. The widget header shows 'Car-0001' and a status indicator. Below the header is a large empty box with the text 'This is the beginning of your interactions...'. At the bottom of the widget, there is a dropdown menu for 'Driver Mobile' with 'Driver' selected. A tooltip above the dropdown says 'Use this to send the driver latest updates'. The bottom right corner of the widget area says 'Powered by: VONAGE'.

# Sending Messages from the UI

## Sending an SMS or WhatsApp Message

To send a message, navigate to the record for the recipient of that message. You will see this UI on the page:



The screenshot shows the Vonage Communication Platform interface for sending a message to David Monaco. The interface includes a header with the contact name and status, a message history section, and a message composition area. The message history shows a message from David Monaco at 08:43 and a response at 08:44. The composition area has a 'To' field set to 'Phone' (marked with a '2'), a 'From' field (marked with a '1'), and a large text input area (marked with a '3'). Below the input area are buttons for 'Attach', 'Template', and 'Clear'. A 'Send' button is also visible. The message history shows a message from David Monaco at 08:43 and a response at 08:44 (marked with a '4').

**Vonage Communication Platform**

**David Monaco**

07-Oct-2024

David Monaco  
Hello, I would like to book a car service. Reg: AA12XYZ  
08:43

Thank you, will 21st July 8am be suitable for you?  
08:44

To: Phone **2** From: **1**

**3** Attach Template Clear


Powered by: **VONAGE** Part of Ericsson

1. First, choose the brand endpoint you want to use to send the message using the drop down. This is automatically populated with all of the different phone numbers that you can use, across all of the different channels.
  - a. For example, you may have one brand number for communications in the USA and another for communications within the UK. If the permissions have been enabled, you can use the drop down to choose between the numbers; if not, the default number will be selected.
2. Next, choose the phone number that you want to send the message to; the drop down is automatically populated with all phone numbers associated with that record.
3. You can attach an image to the message using the 'Attach' button; see the [next section](#) for more information. You can also choose to use a predefined template for your message by selecting the 'Template' button - the list of available options will change depending on the selected endpoint.
4. Once you've sent a message, you will be able to see it in the conversation history along with the channel used and the message delivery status. The name of the agent that sent the message will be shown, but you can also see it by hovering over the sender icon.

## Sending a Message with a Resource

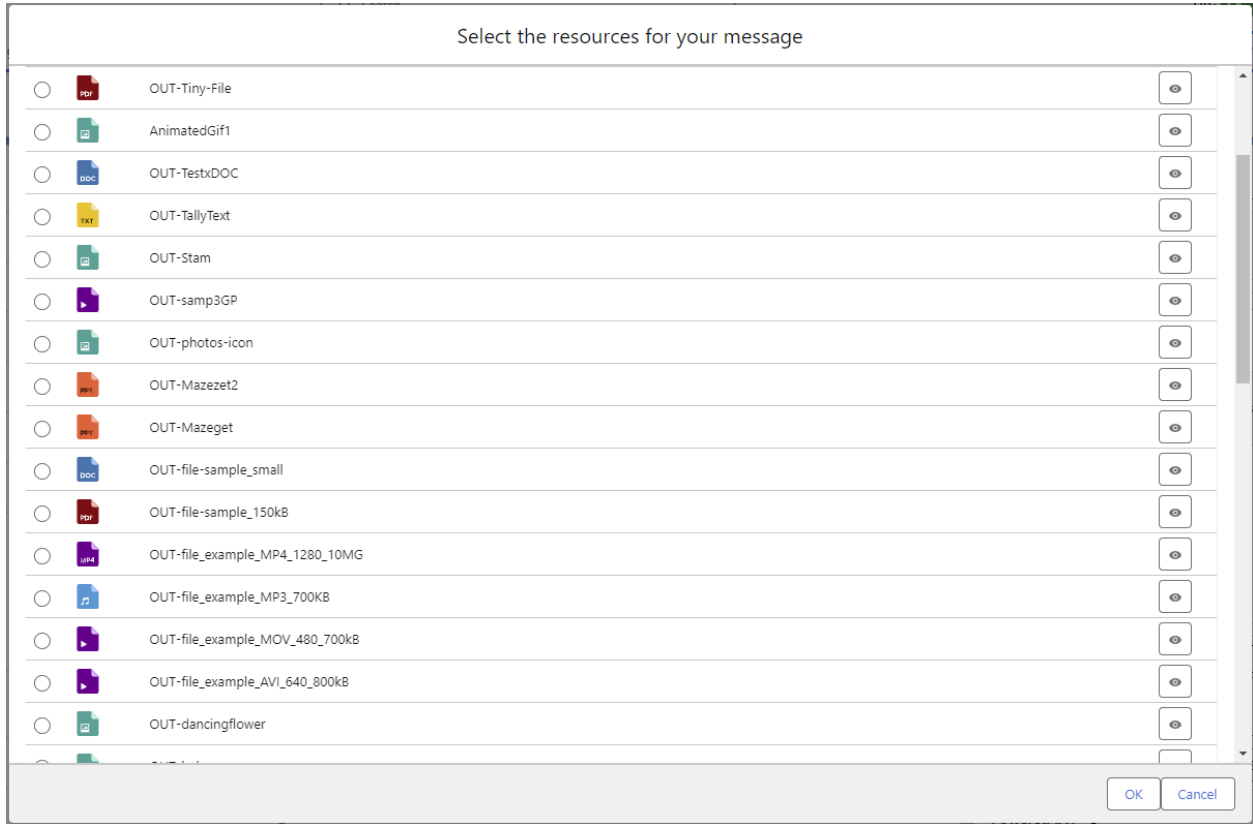
**Note: MMS Images are only supported for USA or Canada numbers. The LVNs used must be approved for [10DLC](#).**

When sending messages, you can attach **one** resource by selecting the 'Attach' button in the bottom left:



The screenshot shows a messaging interface with a 'To:' field set to 'Phone' and a 'From:' field with a dropdown arrow, a speech bubble icon, a UK flag, and the number '44'. Below these fields is a large text input area. At the bottom of the input area, there are three buttons: 'Attach' (with a paperclip icon), 'Template' (with a document icon), and 'Clear' (with a trash can icon). The 'Attach' button is highlighted with a purple rectangular box. At the bottom of the interface, there is a logo that says 'Powered by: VONAGE Part of Ericsson'.

You will see a list of images that you are permitted to send. The icon on the left indicates what type of file that resource is, and you can preview the resource by clicking the eye icon on the right:



These resources are managed through the 'Files' section. Once selected the resource will be presented in the message editor, and once sent it will be shown in the conversation history:



# Flow Development

BOTs can be developed very easily using the Salesforce Flow infrastructure. As the creation of any interaction is published by Platform Event, we can use the **Platform Event-Triggered** Flow in order to build a BOT.

The package includes several example BOTS based on Flow templates:

Flow Template	Feature
Vonage WhatsApp Notification Example	Send a WhatsApp message using a WhatsApp template.
Vonage SMS Notification Example	Send an SMS message using an SMS template.
Vonage Handle Incoming Message Example	Create a new Lead record based on incoming messages from an unknown customer. Assign the new Lead to the designated agent.
Vonage Send Automatic Response Example	Accept and incoming message and response back based on its content

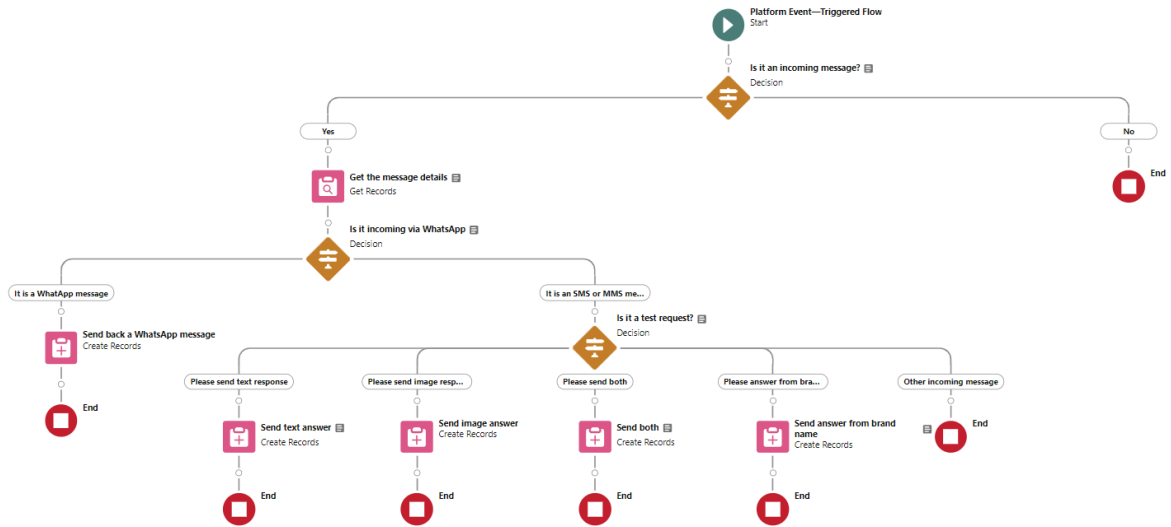
This guide will explain how to use the template to send an automatic response based on an incoming message, but full details on all of the templates can be found in the [Admin & Developer guide](#). All of the provided flow templates are deactivated. In order to use one, please copy it and amend it to the specific org needs.

## Send Automatic Response


**Use case:** Send an automatic response, based on the incoming message content and channel.

**The flow:** The Flow is activated when a new SMS/WhatsApp message is created in org. If it is an outbound message, the flow ends. If it is an inbound message, the flow will do the following:

- Fetch the message details.
- If the message is from a known customer, the flow ends. In this case, the message is already linked to the record (Lead or Case etc) which includes the caller phone number.
- If it is a WhatsApp message - send a response via WhatsApp.
- If it is an SMS message - check the message content. In this example, we respond to messages which include a specific phrase, and ignore any other message.
- Based on the phrase used in the message, a suitable SMS response is sent back.



### The platform event trigger for initiating the Flow:

 Choose Platform Event
✕

The flow subscribes to the specified platform event. When a platform event message is received, the flow is triggered to run.

\* Platform Event

InteractionEvent

[Advanced Settings](#)

**Run this flow as:** ⓘ

- User that triggered the event
- Default Workflow User [Tally..Tsabary](#)

### Is it an incoming message?

For the "Yes" section:

Edit Decision
✕

Is it an incoming message? (*Is\_incoming\_message*)

Verify whether the Platform Event is related to an incoming message or an outgoing message.

---

**Outcomes** For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

---

OUTCOME ORDER	OUTCOME DETAILS
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>⋮</span> <span>Yes</span> <span>+</span> </div>	<p>* Label <input type="text" value="Yes"/></p> <p>* Outcome API Name <input type="text" value="yes_it_is_incoming_message"/></p> <p>Condition Requirements to Execute Outcome</p> <p style="border: 1px solid #ccc; padding: 2px;">All Conditions Are Met (AND)</p> <p>Resource: <input type="text" value="\$Record &gt; Interaction Direction"/> Operator: <input type="text" value="Equals"/> Value: <input type="text" value="Inbound"/></p> <p style="text-align: right;"></p> <p><a href="#">+ Add Condition</a></p>
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>⋮</span> <span>No</span> </div>	

**Note: While copying the template to your own Flow, make sure that the relevant field names are updated to reflect the package name space.**

For the "No" section:

Edit Decision
✕

Is it an incoming message? (*Is\_incoming\_message*)

Verify whether the Platform Event is related to an incoming message or an outgoing message.


---


**Outcomes** For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

---

OUTCOME ORDER	OUTCOME DETAILS
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>⋮</span> <span>Yes</span> </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>⋮</span> <span>No</span> </div>	<p>* Label <input type="text" value="No"/></p>

## Get the message details:

 **Edit Get Records** ×

Get the message details (*Get\_the\_message\_details*) 

Get the message details.  
The message Id is provided in the Platform Event

---

### Get Records of This Object


\* Object

---

### Filter Interaction Records

Condition Requirements

All Conditions Are Met (AND) ▼


Field	Operator	Value	
<input type="text" value="Id"/>	<input type="text" value="Equals"/> ▼	<input type="text" value="\$Record &gt; InteractionId"/> ✕	

[+ Add Condition](#)

---

### Sort Interaction Records

Sort Order

▼  If you store only the first record, filter by a unique field, such as ID.

---

### How Many Records to Store

Only the first record

All records


### How to Store Record Data


Automatically store all fields

Choose fields and let Salesforce do the rest

Choose fields and assign variables (advanced)

## Is the message incoming via WhatsApp?

 Edit Decision
×

**Is it incoming via WhatsApp** (*Is\_it\_incoming\_via\_WhatsApp*) 

Is the incoming request arriving via WhatsApp channel


---


**Outcomes** For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER	OUTCOME DETAILS	
1 <span style="font-size: 20px;">+</span>		
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 20px;">⋮</span> It is a WhatsApp message                             </div> <div style="border: 1px solid #ccc; padding: 5px;">                                 It is an SMS or MMS message                             </div>	<p><small>* Label</small></p> <input style="width: 100%;" type="text" value="It is a WhatsApp message"/>	<p><small>* Outcome API Name</small></p> <input style="width: 100%;" type="text" value="It_is_a_WhatsApp_message"/>
	<p><small>Condition Requirements to Execute Outcome</small></p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;"> <span>All Conditions Are Met (AND)</span> </div>	
	<p><small>Resource</small></p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <span style="font-size: 20px;">⋮</span> Interaction from Get_the_messa... <span style="font-size: 18px;">×</span> </div>	<p><small>Operator</small></p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <span>Equals</span> <span style="font-size: 20px;">▾</span> </div>
	<p><small>Value</small></p> <input style="width: 100%;" type="text" value="WhatsApp"/>	
	<span style="border: 1px solid #ccc; padding: 2px 10px;">+ Add Condition</span>	

Note: The Resource name is: `{!Get_the_message_details.VC_Channel_c}`

## Is the message incoming via SMS/MMS?

 Edit Decision
×

**Is it incoming via WhatsApp** (*Is\_it\_incoming\_via\_WhatsApp*) 

Is the incoming request arriving via WhatsApp channel

---

**Outcomes** For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER	OUTCOME DETAILS	
1 <span style="font-size: 20px;">+</span>		
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 20px;">⋮</span> It is a WhatsApp message                             </div> <div style="border: 1px solid #ccc; padding: 5px;">                                 It is an SMS or MMS message                             </div>	<p><small>* Label</small></p> <input style="width: 100%;" type="text" value="It is an SMS or MMS message"/>	



## Send back a WhatsApp message

✕

Edit Create Records

Send back a WhatsApp message (*Send\_back\_a\_WhatsApp\_message*)

**How Many Records to Create**

One

Multiple

**How to Set the Record Fields**

Use all values from a record

Use separate resources, and literal values

**Create a Record of This Object**

\* Object

Interaction


**Set Field Values for the Interaction**

Field	Value	
<div style="border: 1px solid #ccc; padding: 2px;">VC_Brand_Endpoint__c</div>	← <div style="border: 1px solid #ccc; padding: 2px;">Interaction from Get_the_message_details &gt; Bra... X</div>	
<div style="border: 1px solid #ccc; padding: 2px;">VC_Content_Text__c</div>	← <div style="border: 1px solid #ccc; padding: 2px;">hello from the flow to WA user</div>	
<div style="border: 1px solid #ccc; padding: 2px;">VC_Direction__c</div>	← <div style="border: 1px solid #ccc; padding: 2px;">Outbound</div>	
<div style="border: 1px solid #ccc; padding: 2px;">VC_Interaction_Media__c</div>	← <div style="border: 1px solid #ccc; padding: 2px;">Text</div>	
<div style="border: 1px solid #ccc; padding: 2px;">VC_Party_Endpoint__c</div>	← <div style="border: 1px solid #ccc; padding: 2px;">Interaction from Get_the_message_details &gt; Par... X</div>	
<div style="border: 1px solid #ccc; padding: 2px;">VC_Thread__c</div>	← <div style="border: 1px solid #ccc; padding: 2px;">Interaction from Get_the_message_details &gt; Thr... X</div>	

+ Add field

### Is it a test request?

Filtering the Flow for different types of responses based on the message content. The message content (the Resource field) is: `{!Get_the_message_details.VC_Content_Text__c}`


**Edit Decision** ✕  
 Is it a test request? (*Is\_it\_a\_test\_request*)   
 Parse the message text

**Outcomes** For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.


OUTCOME ORDER	OUTCOME DETAILS		Delete Outcome
<ul style="list-style-type: none"> <li>⋮ Please send text response</li> <li>⋮ Please send image response</li> <li>⋮ Please send both</li> <li>⋮ Please answer from brand name</li> <li>Other incoming message</li> </ul>	* Label <input type="text" value="Please send text response"/>	* Outcome API Name <input type="text" value="Please_send_text_response"/>	
	Condition Requirements to Execute Outcome <input type="text" value="All Conditions Are Met (AND)"/>		
	Resource <input type="text" value="A Interaction from Get_the_messa..."/>	Operator <input type="text" value="Contains"/>	Value <input type="text" value="Please send text"/>
	<input type="button" value="+ Add Condition"/>		



**Edit Decision** ✕  
 Is it a test request? (*Is\_it\_a\_test\_request*)   
 Parse the message text

**Outcomes** For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER	OUTCOME DETAILS		Delete Outcome
<ul style="list-style-type: none"> <li>⋮ Please send text response</li> <li>⋮ Please send image response</li> <li>⋮ Please send both</li> <li>⋮ Please answer from brand name</li> <li>Other incoming message</li> </ul>	* Label <input type="text" value="Please send image response"/>	* Outcome API Name <input type="text" value="Please_send_image_response"/>	
	Condition Requirements to Execute Outcome <input type="text" value="All Conditions Are Met (AND)"/>		
	Resource <input type="text" value="A Interaction from Get_the_messa..."/>	Operator <input type="text" value="Contains"/>	Value <input type="text" value="Please send image"/>
	<input type="button" value="+ Add Condition"/>		

## Send text response

 Edit Create Records
✕

Send text answer (*Send\_text\_answer*) 

Create an outgoing response which includes only text message

**How to Set the Record Fields**

Use all values from a record









Use separate resources, and literal values

**Create a Record of This Object**

\*Object

Interaction

**Set Field Values for the Interaction**

Field	Value	
VC__Brand_Endpoint__c	Interaction from Get_the_message_details > Br... ✕	← 
VC__Channel__c	SMS	← 
VC__Content_Text__c	Hello from Vonage BOT example	← 
VC__Direction__c	Outbound	← 
VC__Interaction_Originator_Type__c	Bot	← 
VC__Interaction_Originator__c	Vonage BOT example	← 
VC__Party_Endpoint__c	Interaction from Get_the_message_details > Pa... ✕	← 
VC__Thread__c	Interaction from Get_the_message_details > Th... ✕	← 

+ Add Field

## Send image response

✕

📄
Edit Create Records

Send image answer (*Send\_image\_answer*)
✎

🔵
Multiple

**How to Set the Record Fields**

Use all values from a record

Use separate resources, and literal values

**Create a Record of This Object**

\*Object

Interaction

**Set Field Values for the Interaction**

Field	←	Value	
<div style="border: 1px solid #ccc; padding: 5px;">VC_Brand_Endpoint_c</div>		<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> <span style="font-size: 0.8em; margin-right: 5px;">Aa</span> <span>Interaction from Get_the_message_details &gt; Br...</span> <span style="margin-left: 5px; border: 1px solid #ccc; padding: 0 2px;">✕</span> </div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>
<div style="border: 1px solid #ccc; padding: 5px;">VC_Channel_c</div>		<div style="border: 1px solid #ccc; padding: 5px;">MMS</div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>
<div style="border: 1px solid #ccc; padding: 5px;">VC_Direction_c</div>		<div style="border: 1px solid #ccc; padding: 5px;">Outbound</div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>
<div style="border: 1px solid #ccc; padding: 5px;">VC_Interaction_Originator_Type_c</div>		<div style="border: 1px solid #ccc; padding: 5px;">Bot</div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>
<div style="border: 1px solid #ccc; padding: 5px;">VC_Interaction_Originator_c</div>		<div style="border: 1px solid #ccc; padding: 5px;">Vonage BOT example</div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>
<div style="border: 1px solid #ccc; padding: 5px;">VC_Party_Endpoint_c</div>		<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> <span style="font-size: 0.8em; margin-right: 5px;">Aa</span> <span>Interaction from Get_the_message_details &gt; Pa...</span> <span style="margin-left: 5px; border: 1px solid #ccc; padding: 0 2px;">✕</span> </div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>
<div style="border: 1px solid #ccc; padding: 5px;">VC_Resources_c</div>		<div style="border: 1px solid #ccc; padding: 5px;">0698d0000F4grxAAB</div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>
<div style="border: 1px solid #ccc; padding: 5px;">VC_Thread_c</div>		<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> <span style="font-size: 0.8em; margin-right: 5px;">Aa</span> <span>Interaction from Get_the_message_details &gt; Th...</span> <span style="margin-left: 5px; border: 1px solid #ccc; padding: 0 2px;">✕</span> </div>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>

+ Add Field

Note: MMS images are only supported in the USA and Canada. The **VC\_Resources\_c** value is the record Id of the ContentDocument that includes the image, which can be found in the Files app.

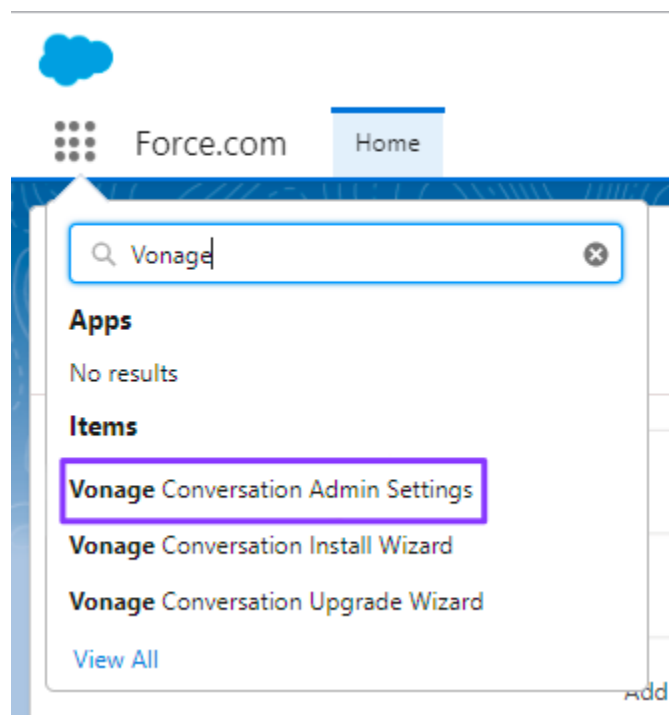


## Troubleshooting and Support

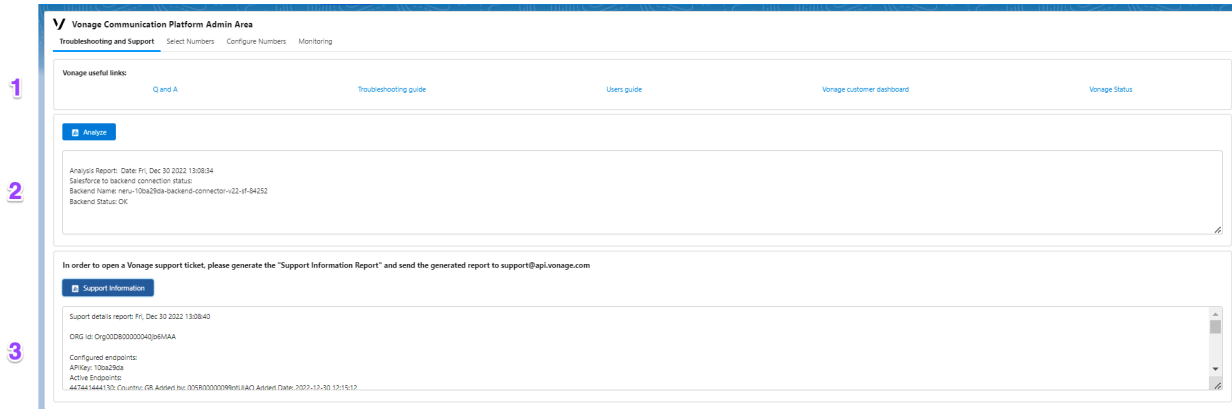
The Admin Configuration Page allows the admin to access several useful troubleshooting and support functions:

1. Check the backend connectivity status.
2. Fetch logs.
3. See shortcuts to the Vonage documentation, status page and customer dashboard.

To access the page search Vonage to find the 'Vonage Conversation Admin Settings':



The 'Troubleshooting and Support' tab contains three sections:



## Vonage Useful links

Links to the Vonage documentation, status page and customer dashboard.

## Analyze

Clicking the 'Analyze' button will fetch the latest status of the connection between the ORG and the Vonage backend connector. You will see the name of the backend connector instance, along with the backend status. This information can be used for further communications with [Vonage support](#) in case of connectivity issues - any tickets raised should be under the category 'Using Conversations for Salesforce'.

## Support Information

Clicking the 'Support Information' button will fetch the latest connection status, current configuration details and logs in one report. This information can be used for troubleshooting and further communications with the Vonage [support team](#) - any tickets raised should be under the category 'Using Conversations for Salesforce'.