

# 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 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.

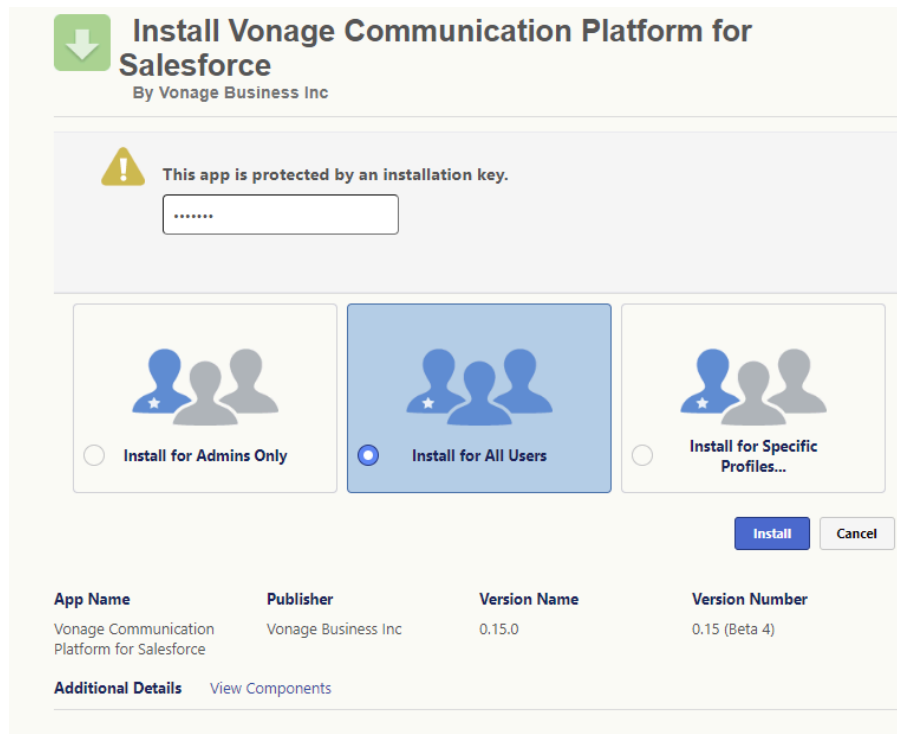
1. A [Vonage account](#) with your **api-key** and at least one [purchased long virtual number](#) (LVN).

2. The LVN should be configured to use SMS/MMS.
  - a. 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](#).

## 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:




The screenshot shows the Salesforce app installation interface. At the top, it says "Install Vonage Communication Platform for Salesforce" by Vonage Business Inc. Below this is a warning message: "This app is protected by an installation key." with a text input field containing six asterisks. There are three radio button options for installation scope: "Install for Admins Only", "Install for All Users" (which is selected), and "Install for Specific Profiles...". At the bottom right are "Install" and "Cancel" buttons. Below the options is a table with the following data:

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

Below the table, there is a link for "Additional Details" and a button for "View Components".

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.

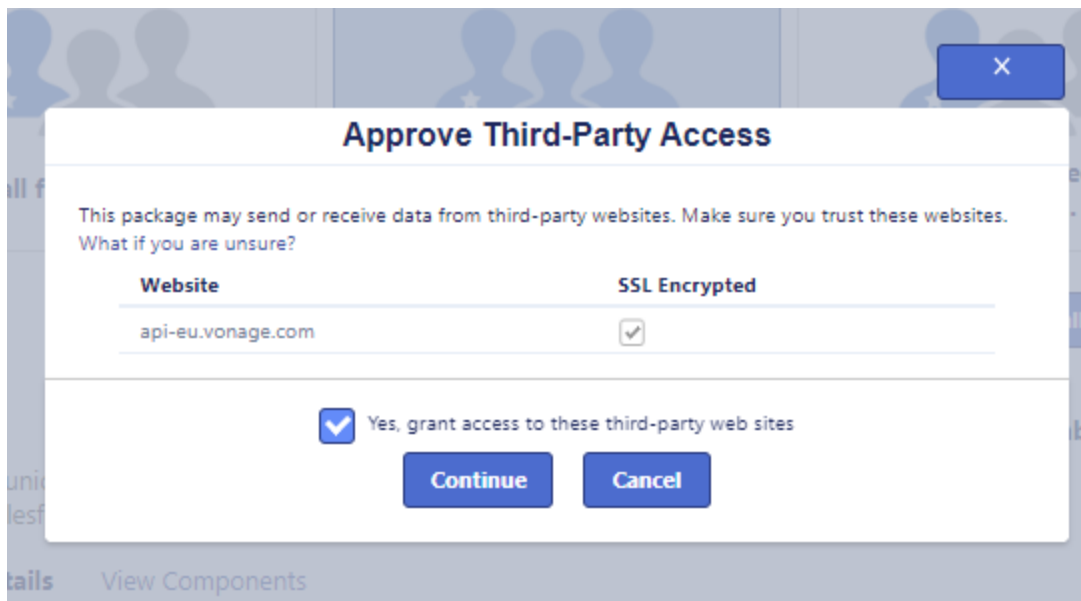


**This app is taking a long time to install.**  
You will receive an email after the installation has completed.

---

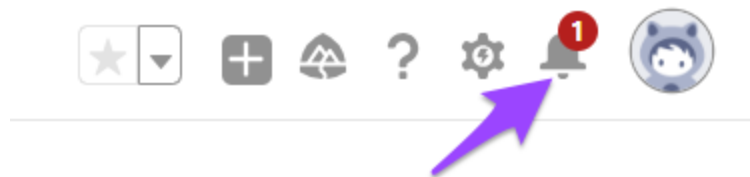
App Name	Publisher	Version Name	Version Number
Vonage Communication Platform for Salesforce	Vonage Business Inc	0.15.0	0.15 (Beta 4)

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

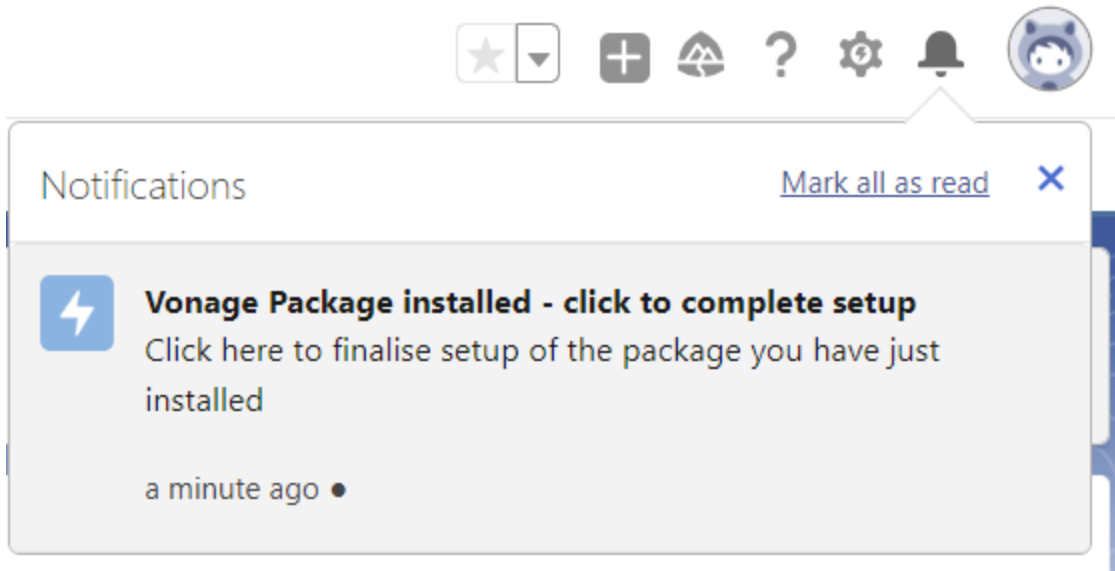


## First Time Deployment

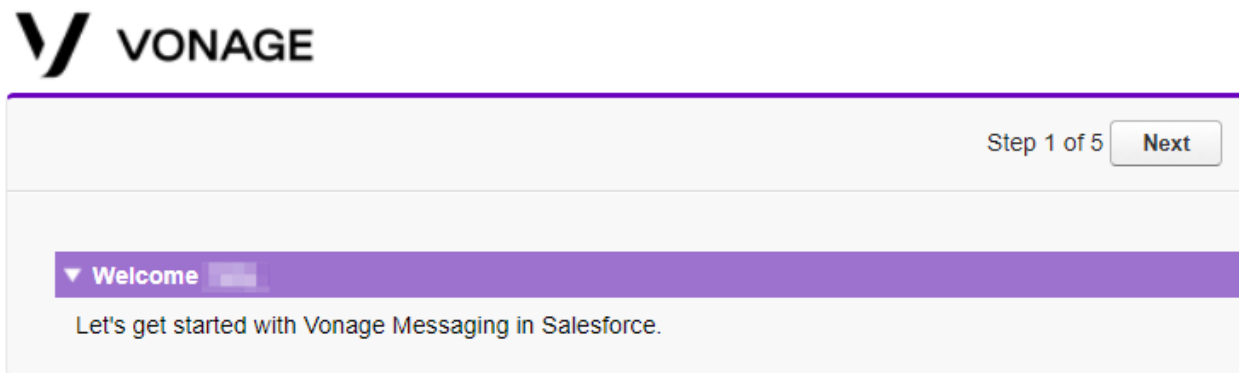
Once the initial deployment is done, the 'Bell Notification' will indicate a new message:



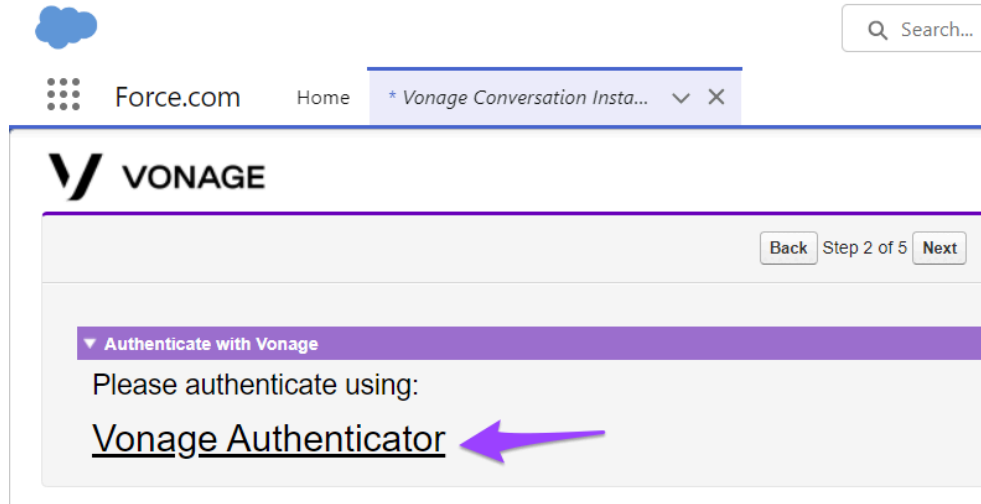
Select the link in the message to complete the setup:



Click 'Next' to start the deployment process:

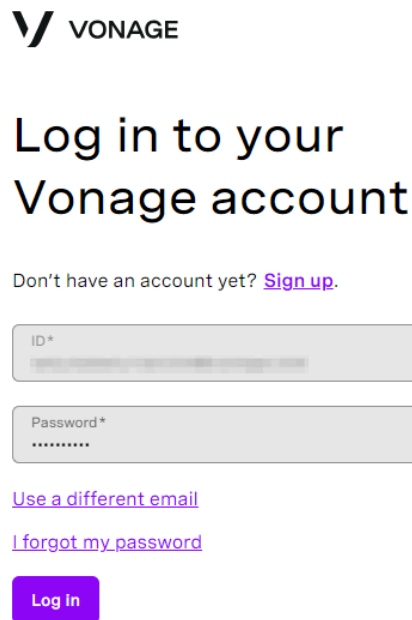


For Step 2, click the link to authenticate with Vonage - please ensure that you are logged-off from any other Salesforce ORG while completing this process:

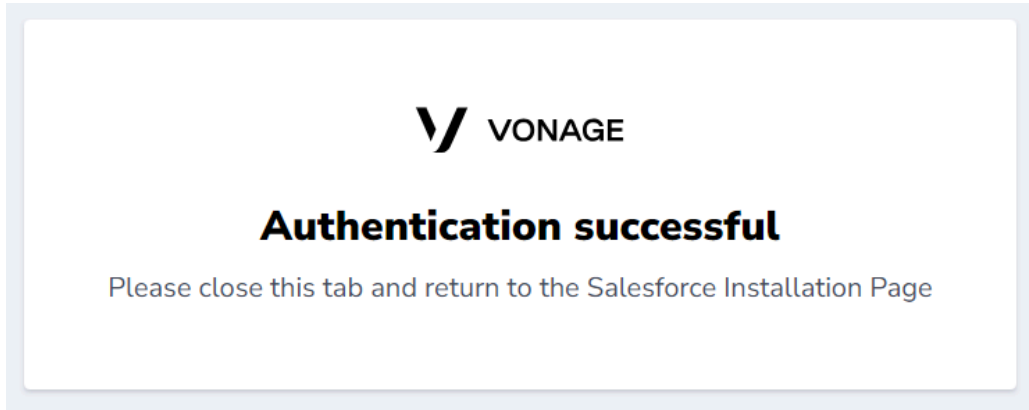


**Failure to complete the authentication will cause errors in later stages of the wizard.**

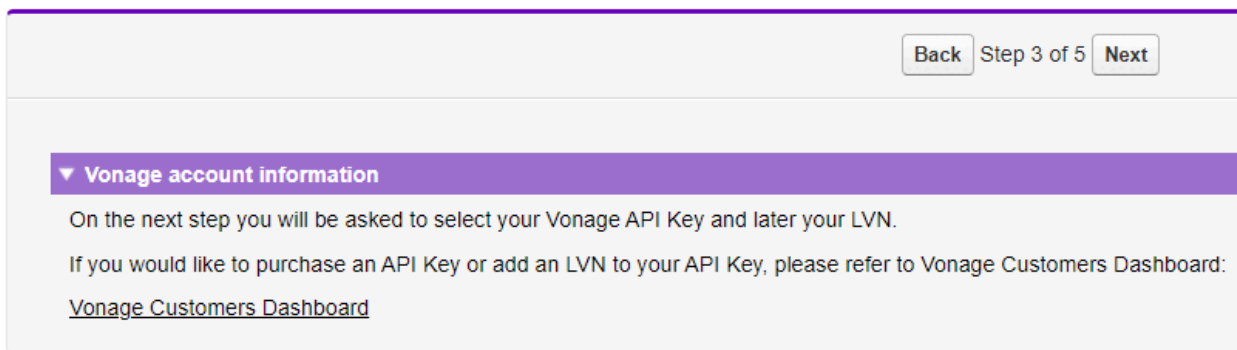
Next, you will be taken to the Vonage customer dashboard to login:



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:



Back Step 4 of 5 Next

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 show the LVNs associated with that API key, along with the channels those numbers support. Select a number, then click Finish:



Back Step 5 of 5 Finish

Number Selection

Please select one of Vonage APIKey [redacted] numbers to use, then click the **Finish** button to complete the setup. This is a long process. Please do not close or refresh this window before the operation is completed.

- 1 [redacted] - Country: US, supported channels: MMS;SMS
- 1 [redacted] - Country: US, supported channels: MMS;SMS
- 447 [redacted] - Country: GB, supported channels: SMS
- 447 [redacted] - Country: GB, supported channels: SMS;WHATSAPP, WABA: [redacted]
- 447 [redacted] - Country: GB, supported channels: SMS

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

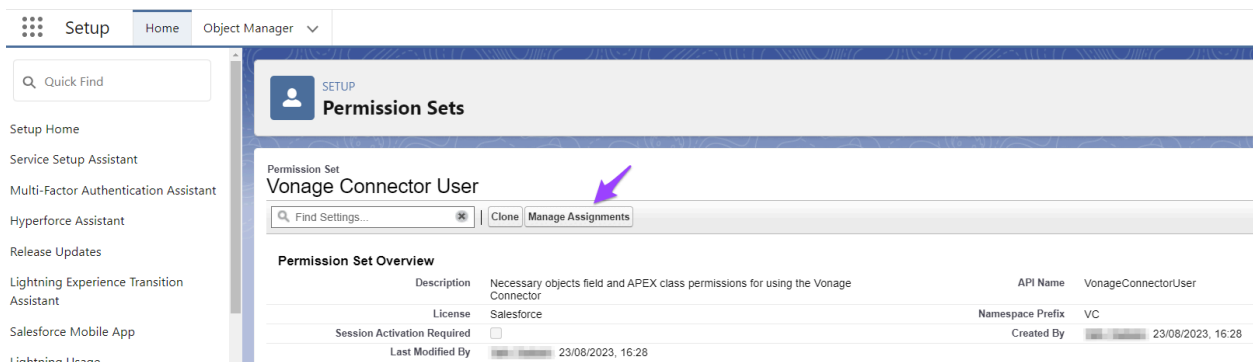
**\*\* Release 0.13 \*\***

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.

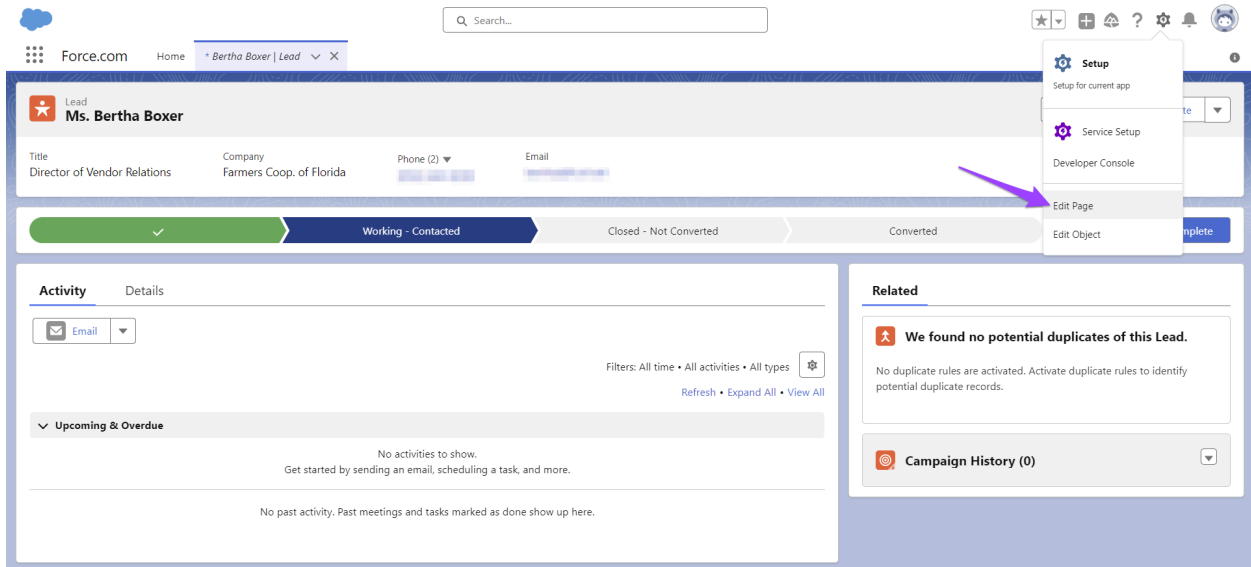
The next section will explain how to add the UI widget to a record page.

# UI Setup

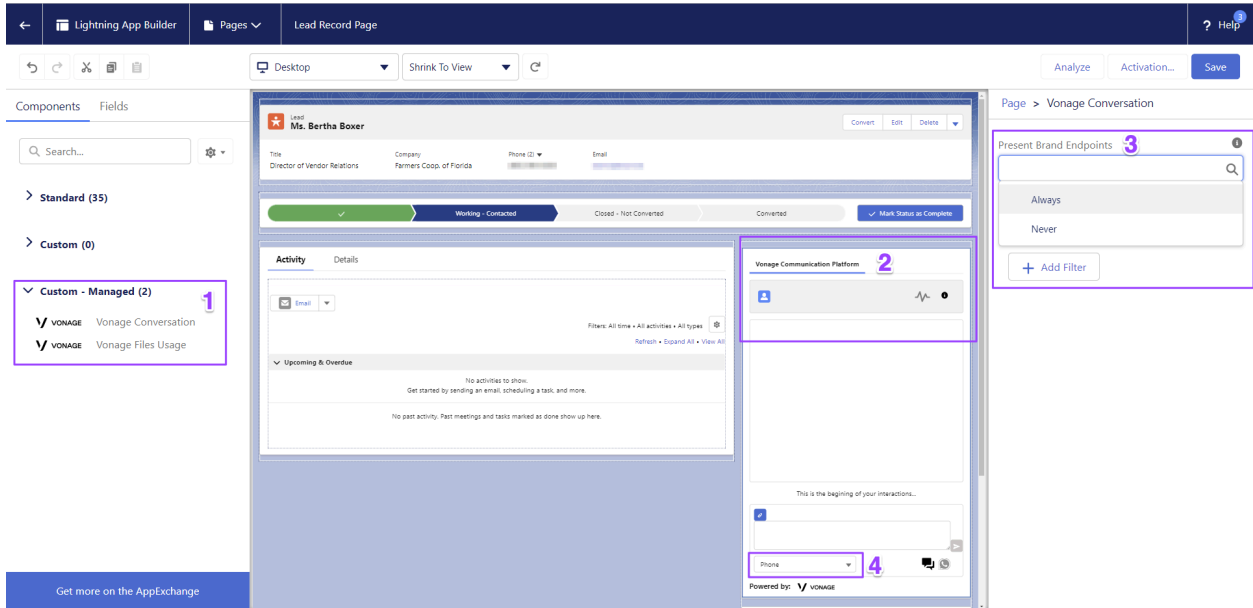
The UI Widget can be added to the record page of any standard sObject or custom sObject.

## Add the widget to a standard sObject

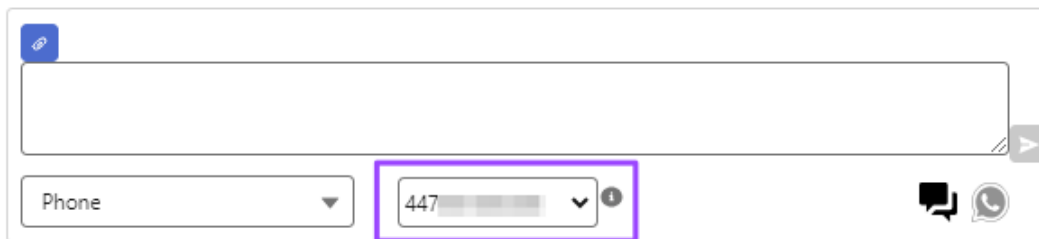
To add the widget to a standard sObject, go to the record page and select "edit page":



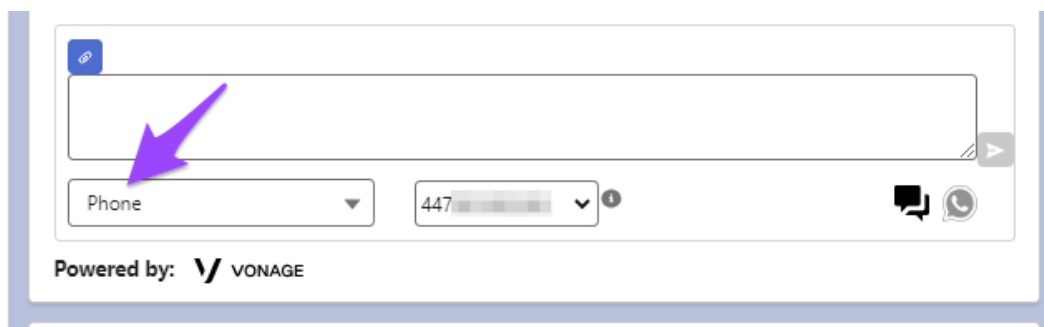
Drag the widget to the suitable place on the page. For example, editing the page for a lead would take you to this screen:



1. Locate the 'Vonage Conversation' UI widget in the Components menu.
2. Drag it to its designated location on the screen.
3. Set if the agent can select the Brand Endpoint. If **Never** - the agent cannot select the Brand Endpoint and the default one will be used automatically. If **Always** - the agent will be able to choose the Brand Endpoint from the drop-down list (4):

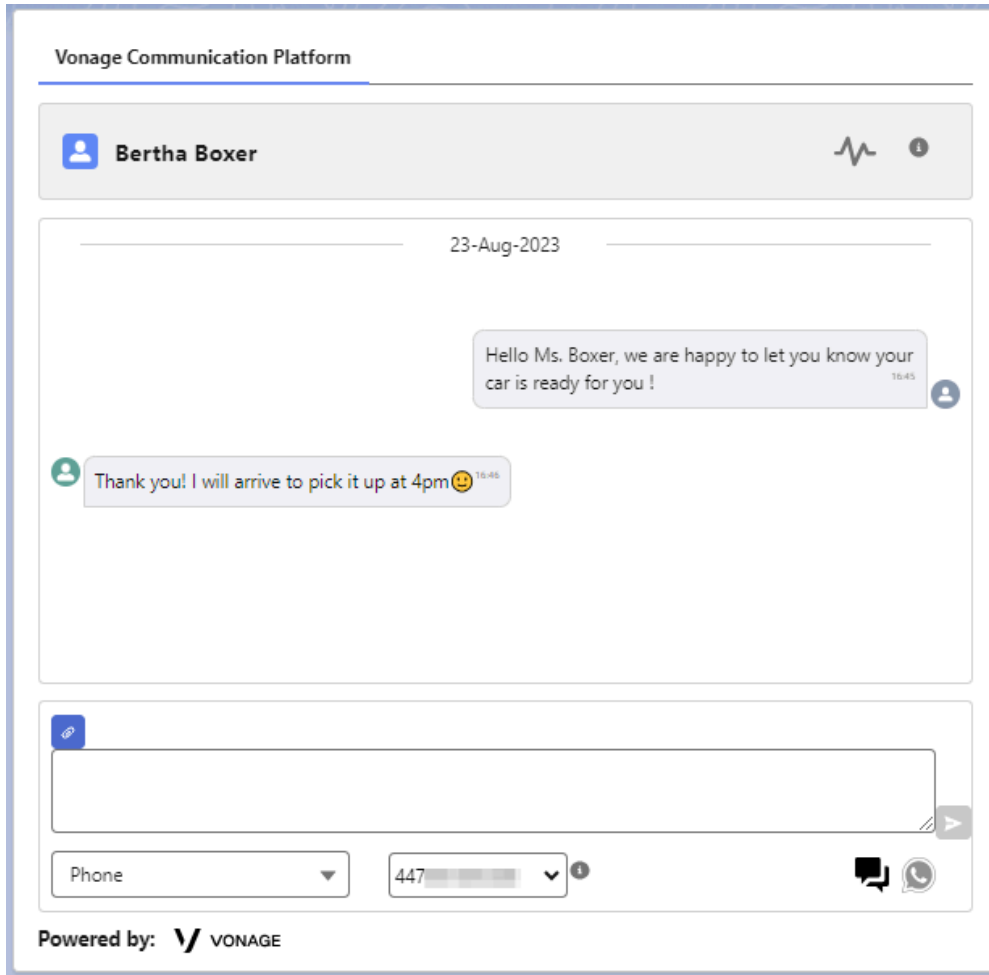


The widget will automatically identify all the populated "Phone" type fields in the record and will add them to the drop down to allow the agent to select which number to send messages to:



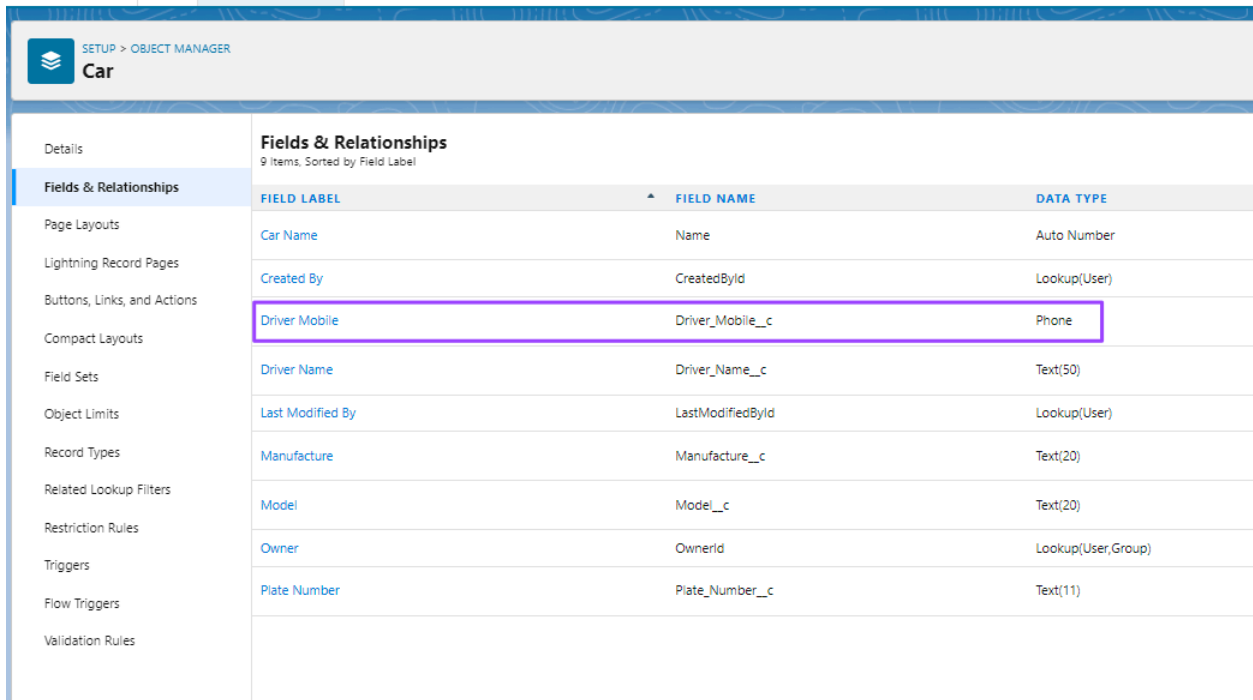
Note that the drop down uses the field label and not the phone number itself in order to allow the admin to hide the original phone numbers if they wish to do so.

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:



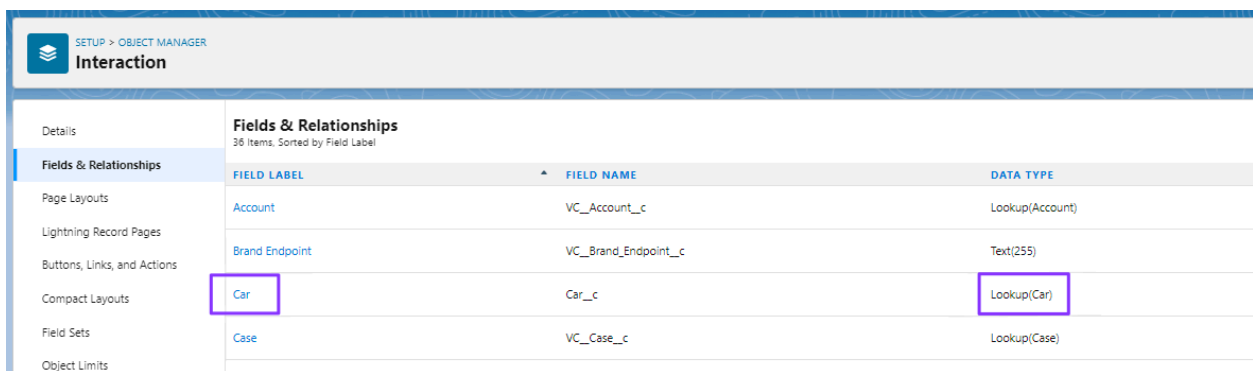
## Add the widget to a custom sObject

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:



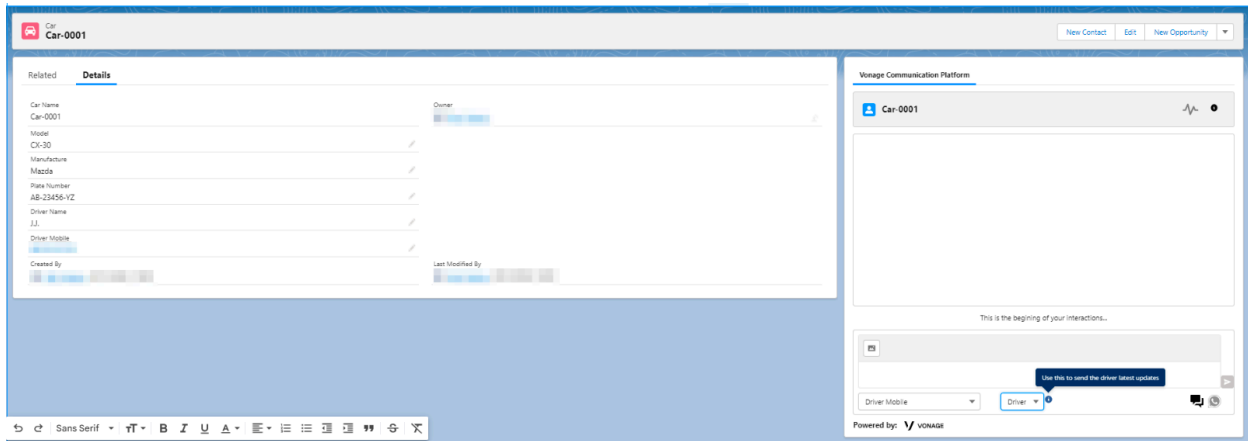
FIELD LABEL	FIELD NAME	DATA TYPE
Car Name	Name	Auto Number
Created By	CreatedById	Lookup(User)
Driver Mobile	Driver_Mobile__c	Phone
Driver Name	Driver_Name__c	Text(50)
Last Modified By	LastModifiedById	Lookup(User)
Manufacture	Manufacture__c	Text(20)
Model	Model__c	Text(20)
Owner	OwnerId	Lookup(User,Group)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

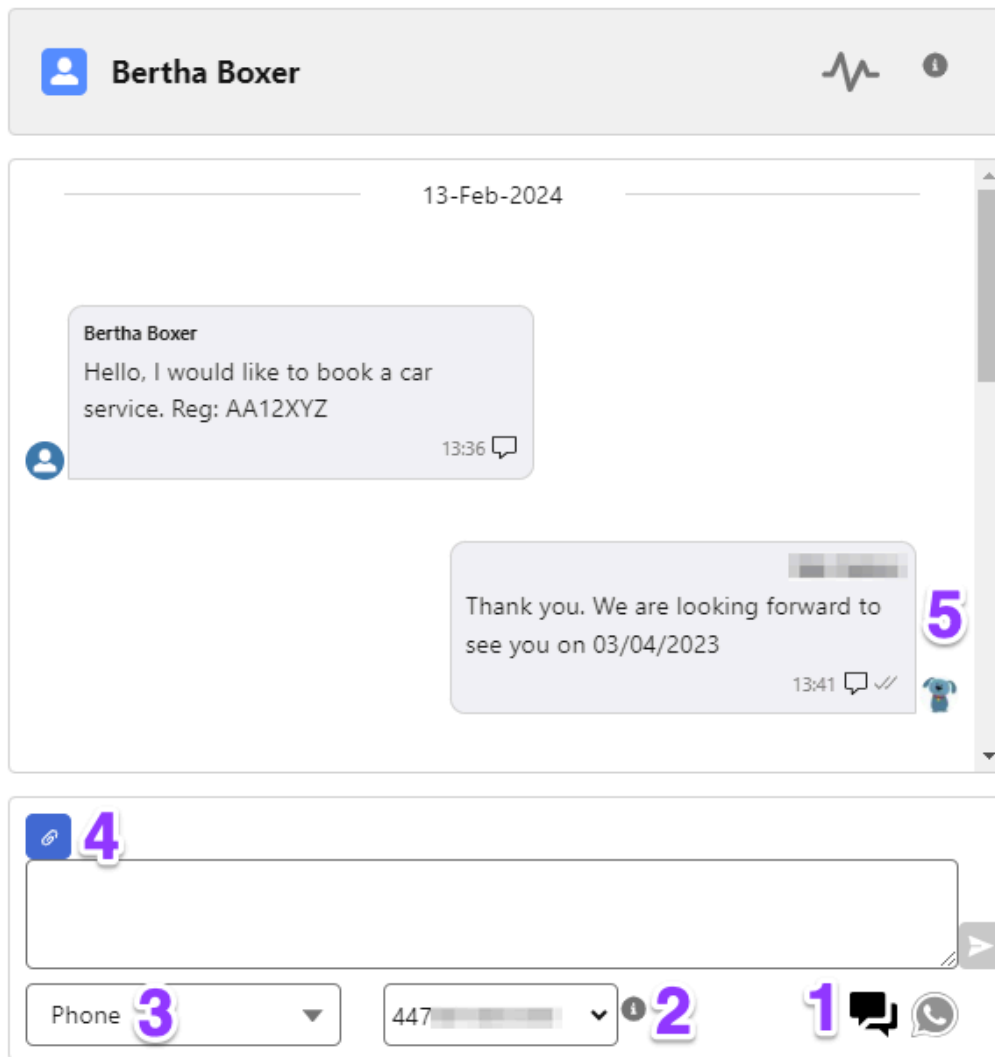
- Car Name: Car-0001
- Model: CX-30
- Manufacture: Mazda
- Plate Number: AB-23456-VZ
- Driver Name: J.J.
- Driver Mobile: [Redacted]
- Created By: [Redacted]
- Last Modified By: [Redacted]

The right sidebar features a 'Vonage Communication Platform' widget for 'Car-0001'. The widget area is currently empty, displaying the text 'This is the beginning of your interactions...'. Below this is a search bar with a dropdown menu set to 'Driver Mobile' and a search button. A tooltip above the search button reads 'Use this to send the driver latest updates'. At the bottom of the widget, it says 'Powered by: VONAGE'.

# Sending Messages from the UI

## Sending an SMS Message

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



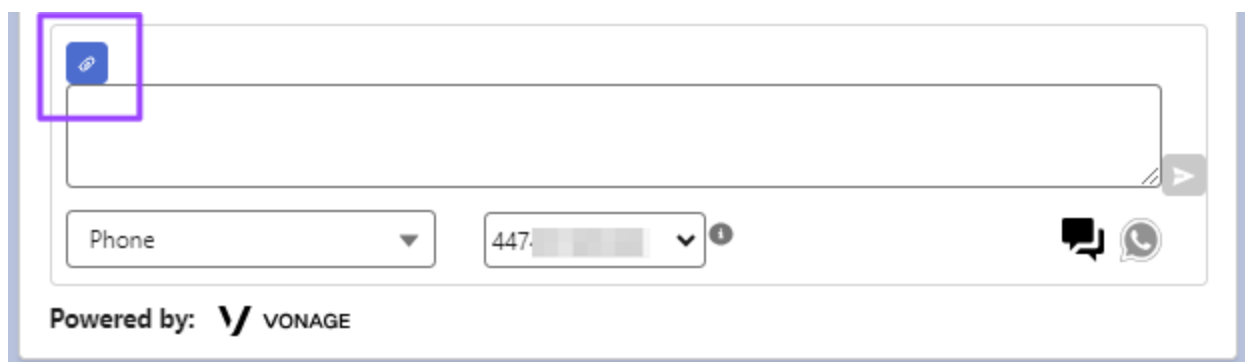


1. First, to send an SMS/MMS message, ensure that the speech bubble icon is selected in the bottom right corner.
2. Next, choose the phone number that you want to send the message to; the drop down is automatically populated with all phone numbers associated with that record.
3. The next drop down shows the brand number the recipient will see. 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.
4. You can attach an image to the message using the link button; see the [next section](#) for more information.
5. 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 link button in the top left:



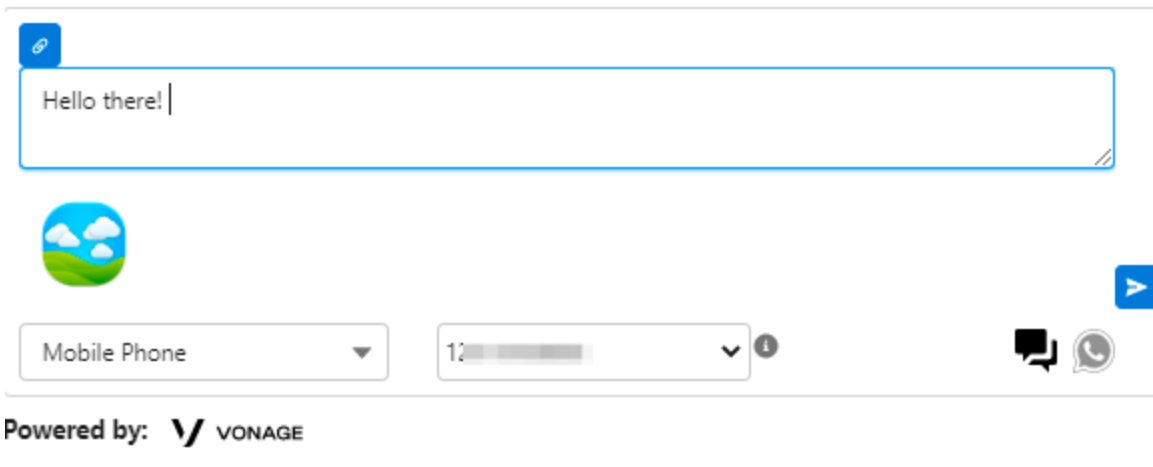
You will see a list of images that you are permitted to send. The icon on the left indicates what type of file that resource is, and you can preview the resource by clicking the eye icon on the right:


Select/Reset the resources for your message

	Label	Description	Preview
<input type="radio"/>	Testx		
<input type="radio"/>	Test		
<input type="radio"/>	sample_640x360		
<input type="radio"/>	sometext		
<input type="radio"/>	app-cloud-mobile-solutions-guide		
<input type="radio"/>	sofa_armchair		
<input type="radio"/>	dog		
<input type="radio"/>	sample_mp4 (1080p)		
<input type="radio"/>	Vonage		
<input type="radio"/>	test		
<input type="radio"/>	Testx		
<input type="radio"/>	Testx		
<input type="radio"/>	Testx		
<input type="radio"/>	rose		

OK Reset Cancel


These resources are managed through the 'Files' section. Once selected the resource will be presented in the message editor; for example, here an image has been attached to the message:




Powered by:  VONAGE

Once sent, it will be shown in the conversation history:





Mobile Phone  1.    

A chat input area with a blue square icon containing a white paper plane symbol on the left. Below it is a long, empty text input field. At the bottom, there are two dropdown menus: the first is labeled "Mobile Phone" and the second contains the number "1." followed by a blurred area. To the right of the dropdowns are two icons: a speech bubble with a checkmark and a telephone handset.

# 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

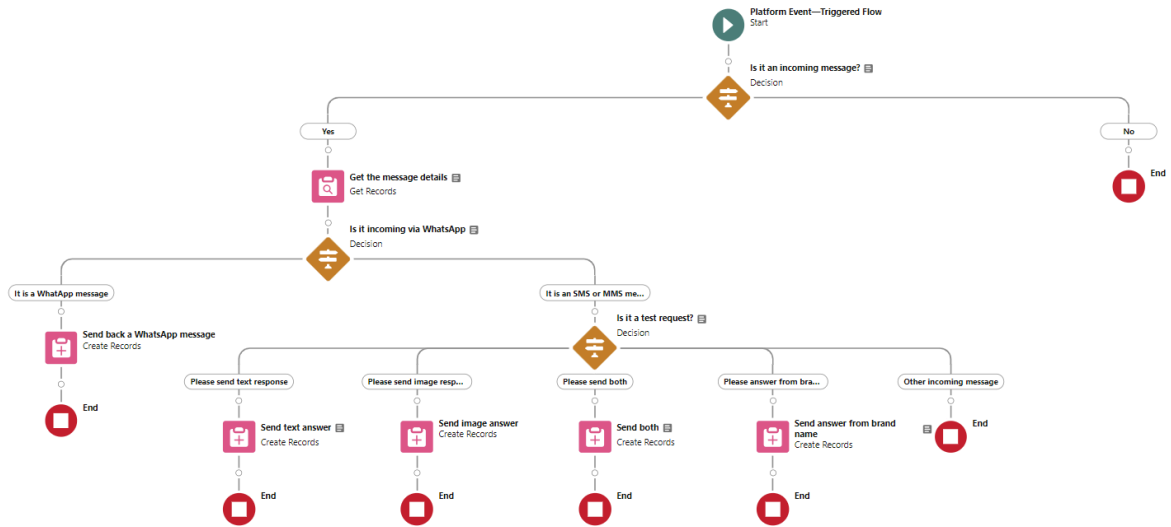
This guide will explain how to use the template to send an automatic response based on an incoming message, but full details on all of the templates can be found in the [Admin & Developer guide](#). All of the provided flow templates are deactivated. In order to use one, please copy it and amend it to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


v [Advanced Settings](#)


**Run this flow as:** i

- 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="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span>⋮</span> Yes           </div> <div style="border: 1px solid #ccc; padding: 5px;">             No           </div>	<div style="margin-bottom: 10px;"> <p><b>* Label</b></p> <input style="width: 100%;" type="text" value="Yes"/> </div> <div style="margin-bottom: 10px;"> <p><b>* Outcome API Name</b></p> <input style="width: 100%;" type="text" value="yes_it_is_incoming_message"/> </div> <p>Condition Requirements to Execute Outcome</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">             All Conditions Are Met (AND)           </div> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%; border-bottom: 1px solid #ccc;">Resource</td> <td style="width: 20%; border-bottom: 1px solid #ccc;">Operator</td> <td style="width: 40%; border-bottom: 1px solid #ccc;">Value</td> </tr> <tr> <td style="border: 1px solid #ccc; padding: 5px;"> <input style="width: 95%;" type="text" value="\$Record &gt; Interaction Direction"/> </td> <td style="border: 1px solid #ccc; padding: 5px;"> <input style="width: 95%;" type="text" value="Equals"/> </td> <td style="border: 1px solid #ccc; padding: 5px;"> <input style="width: 95%;" type="text" value="Inbound"/> </td> </tr> </table> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin-top: 5px;"> <span style="color: #0070C0;">+</span> Add Condition         </div>	Resource	Operator	Value	<input style="width: 95%;" type="text" value="\$Record &gt; Interaction Direction"/>	<input style="width: 95%;" type="text" value="Equals"/>	<input style="width: 95%;" type="text" value="Inbound"/>
Resource	Operator	Value					
<input style="width: 95%;" type="text" value="\$Record &gt; Interaction Direction"/>	<input style="width: 95%;" type="text" value="Equals"/>	<input style="width: 95%;" type="text" value="Inbound"/>					

**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="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span>⋮</span> Yes           </div> <div style="border: 1px solid #ccc; padding: 5px;">             No           </div>	<div style="margin-bottom: 10px;"> <p><b>* Label</b></p> <input style="width: 100%;" type="text" value="No"/> </div>

## 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

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	
+ <span style="font-size: 0.8em;">i</span>		
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 0.8em;">☰</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;">             All Conditions Are Met (AND)           </div>	
	<p><small>Resource</small></p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <span style="font-size: 0.8em;">☰</span> Interaction from Get_the_messa... <span style="margin-left: 5px;">✕</span> </div>	<p><small>Operator</small></p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <span style="font-size: 0.8em;">▼</span> Equals         </div>
	<p><small>Value</small></p> <input style="width: 100%;" type="text" value="WhatsApp"/>	
	<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">+ Add Condition</div> <div style="margin-left: 10px; border: 1px solid #ccc; padding: 2px; font-size: 0.8em;">✕</div>	

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	
+ <span style="font-size: 0.8em;">i</span>		
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 0.8em;">☰</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
VC_Brand_Endpoint__c	Interaction from Get_the_message_details > Bra... <span style="float: right;">✕</span>
VC_Content_Text__c	hello from the flow to WA user
VC_Direction__c	Outbound
VC_Interaction_Media__c	Text
VC_Party_Endpoint__c	Interaction from Get_the_message_details > Par... <span style="float: right;">✕</span>
VC_Thread__c	Interaction from Get_the_message_details > Thr... <span style="float: right;">✕</span>


+ 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 <span>ⓘ</span> <span>+</span>	OUTCOME DETAILS <span>Delete Outcome</span>
<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>	<p>* Label <span style="float: right;">* Outcome API Name</span></p> <p>Please send text response <span style="float: right;">Please_send_text_response</span></p> <p>Condition Requirements to Execute Outcome</p> <p>All Conditions Are Met (AND) <span style="float: right;">▼</span></p> <p>Resource <span style="float: right;">Operator <span style="float: right;">Value</span></span></p> <p> <span style="border: 1px solid gray; padding: 2px;">A Interaction from Get_the_messa... ✕</span> <span style="border: 1px solid gray; padding: 2px;">Contains ▼</span> <span style="border: 1px solid gray; padding: 2px;">Please send text</span> <span style="float: right;"></span> </p> <p><span style="border: 1px solid gray; padding: 2px;">+ Add Condition</span></p>


**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 <span>ⓘ</span> <span>+</span>	OUTCOME DETAILS <span>Delete Outcome</span>
<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>	<p>* Label <span style="float: right;">* Outcome API Name</span></p> <p>Please send image response <span style="float: right;">Please_send_image_response</span></p> <p>Condition Requirements to Execute Outcome</p> <p>All Conditions Are Met (AND) <span style="float: right;">▼</span></p> <p>Resource <span style="float: right;">Operator <span style="float: right;">Value</span></span></p> <p> <span style="border: 1px solid gray; padding: 2px;">A Interaction from Get_the_messa... ✕</span> <span style="border: 1px solid gray; padding: 2px;">Contains ▼</span> <span style="border: 1px solid gray; padding: 2px;">Please send image</span> <span style="float: right;"></span> </p> <p><span style="border: 1px solid gray; padding: 2px;">+ Add Condition</span></p>

## 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
VC_Brand_Endpoint_c	<span style="border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">Aa Interaction from Get_the_message_details &gt; Br... ✕</span> <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</span>
VC_Channel_c	MMS <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</span>
VC_Direction_c	Outbound <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</span>
VC_Interaction_Originator_Type_c	Bot <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</span>
VC_Interaction_Originator_c	Vonage BOT example <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</span>
VC_Party_Endpoint_c	<span style="border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">Aa Interaction from Get_the_message_details &gt; Pa... ✕</span> <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</span>
VC_Resources_c	0698d0000F4grxAAB <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</span>
VC_Thread_c	<span style="border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">Aa Interaction from Get_the_message_details &gt; Th... ✕</span> <span style="float: right; border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px; font-size: 0.8em;">✕</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.

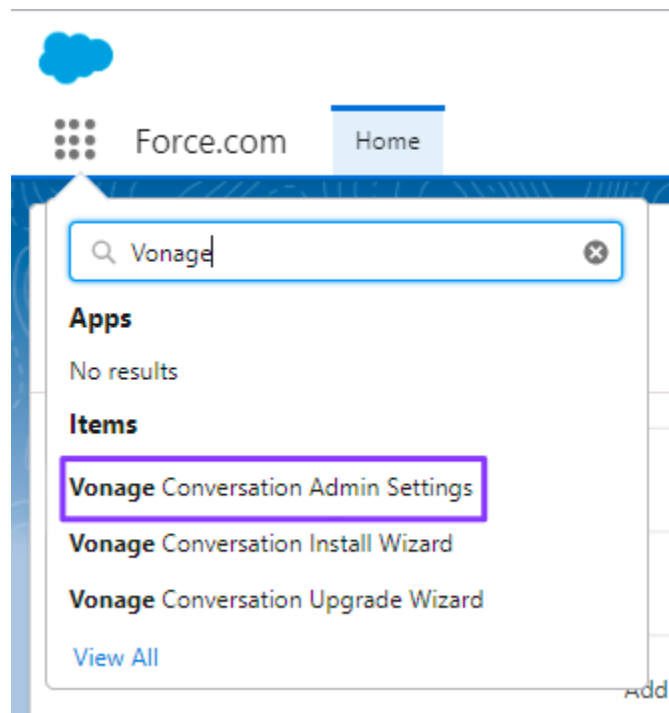


## Troubleshooting and Support

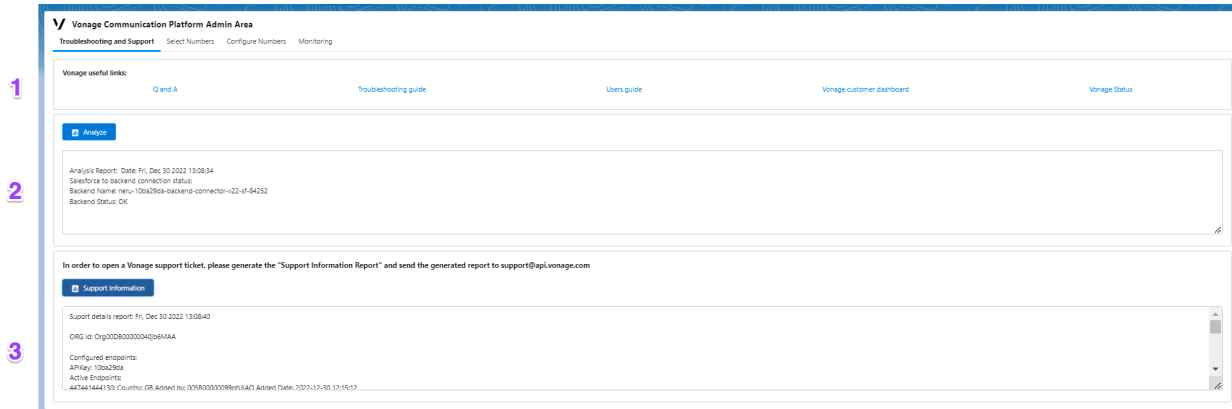
The Admin Configuration Page allows the admin to access several useful troubleshooting and support functions:

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'.