

External Bot Integration

This documentation provides guidelines on how to connect external Bot like Botpress, Flowise, Voiceflow

- [Botpress & WhatsApp Integration](#)
- [Custom Bot integration with vCX](#)

Botpress & WhatsApp Integration



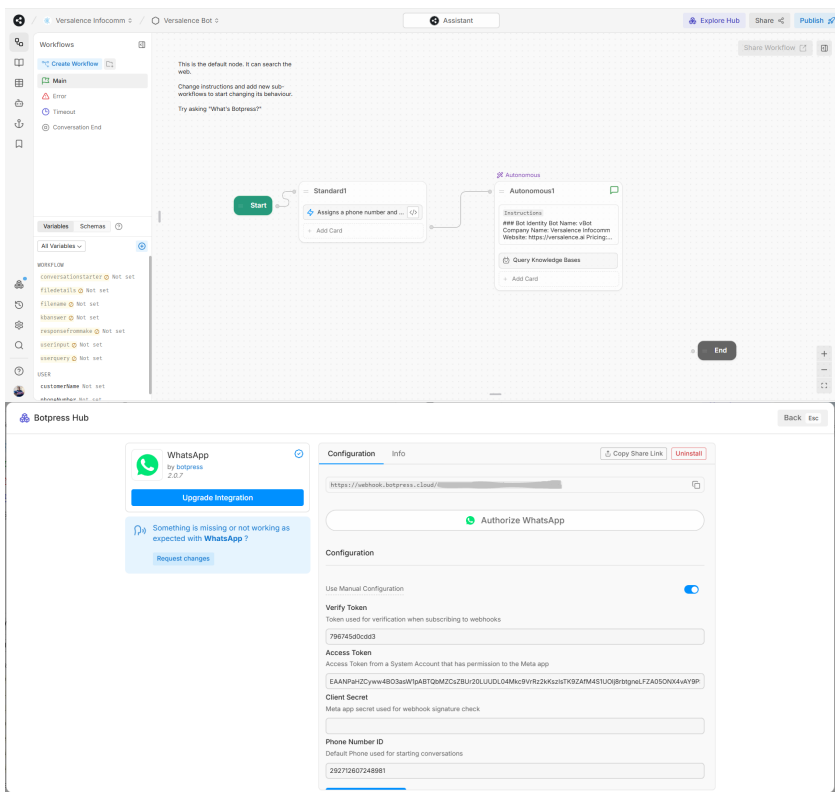
This documentation is based on the integrations possible in the vCX Platform and does not warrant that the external bot studios will work as changes in their architecture are beyond our control. It is our best effort to ensure that the connectivity works

Botpress Setup and WhatsApp Integration

After you have set up your bot in Botpress. Enable the WhatsApp Integration in your Botpress Bot

For vCX to work with Botpress, your WhatsApp Integration has to be in manual mode as shown below. Automated WhatsApp configuration if made, has to be deleted and configured manually

Do not set up Botpress using Authorize WhatsApp!



There are two methods to configure your WhatsApp in vCX

1. Manual Method - Using Facebook App - in <https://developers.facebook.com/apps>
 2. Automated Signup - Using Embedded Signup
- Botpress and vCX support these methods.

For this integration to work, Botpress has to be set up in manual mode

Head to over [vCX](#) and log into your account. Once in your account, click on Settings -> My Plan -> Integrations

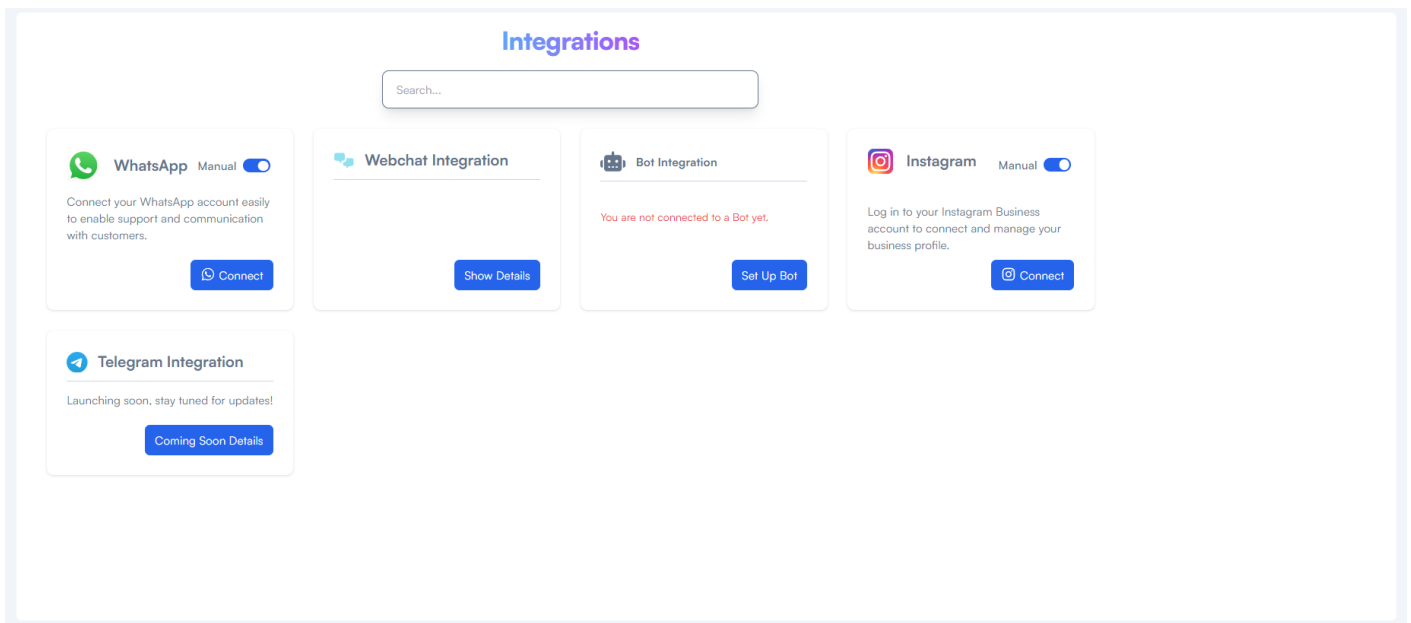
Manual Method

To set it up using manual mode, head over to

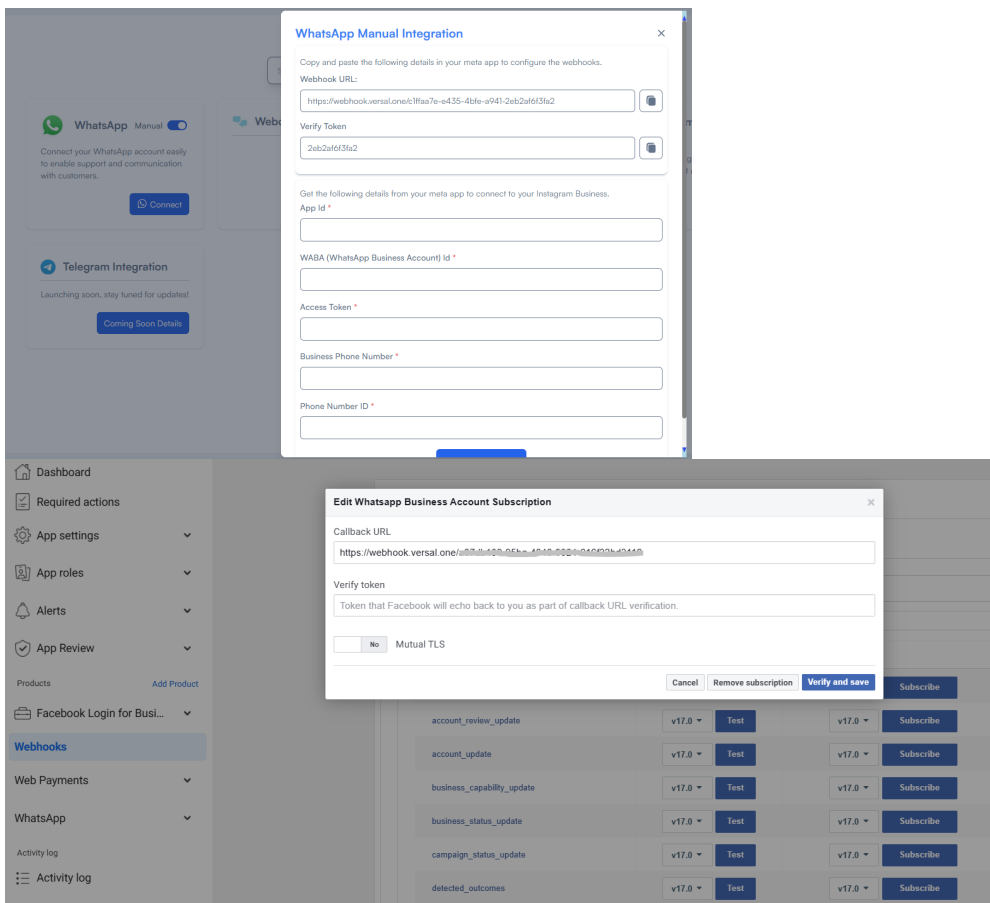
<https://developers.facebook.com/docs/whatsapp/cloud-api/get-started/> OR follow our document [here](#)

Once you have set up your phone number and token secured. Subscribe to vCX Webhook on the developer portal to start receiving incoming messages.

To do so, ensure the Manual option is turned ON in the integration pane for WhatsApp in vCX and click on Connect



From the screen copy the webhook and token and subscribe to Messaging from the META developer port webhook

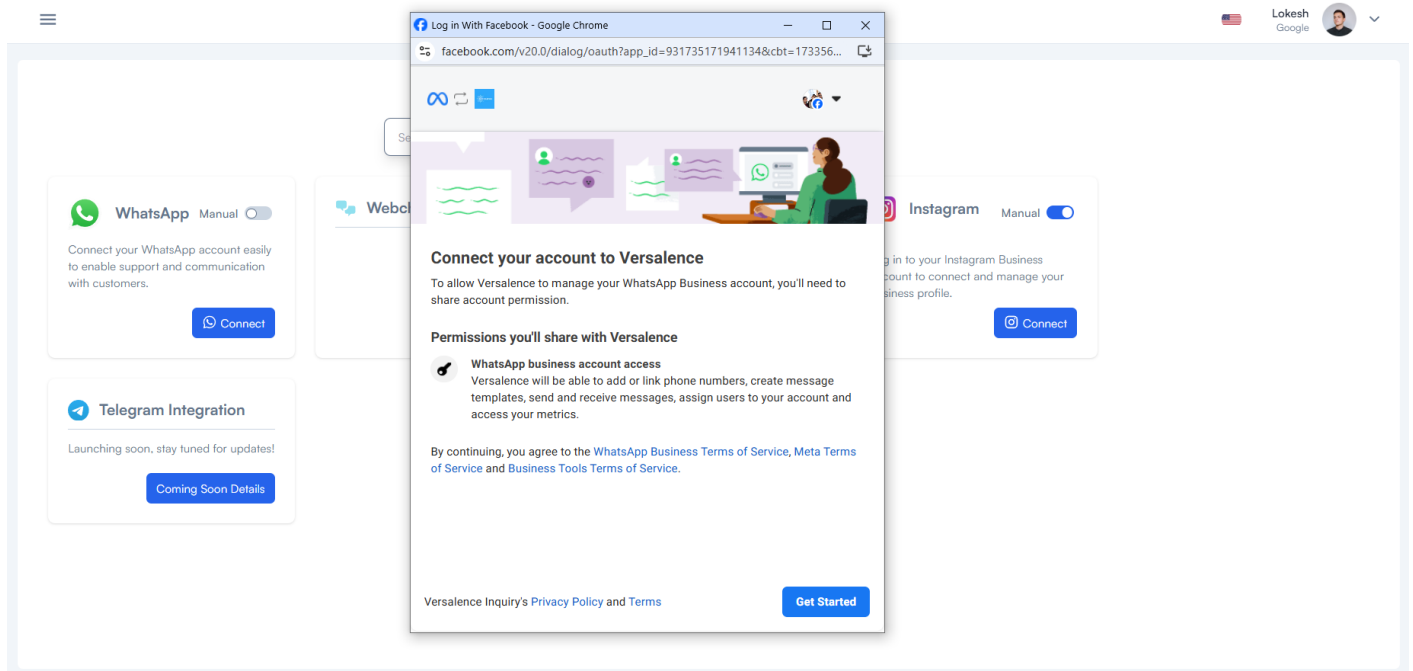


Update all the values from your App and the token in vCX configuration - Do keep them secure as they need to be added to Botpress as well. This completes the setup in vCX for WhatsApp. You can now receive and send chats in vCX

Automated Embedded Signup

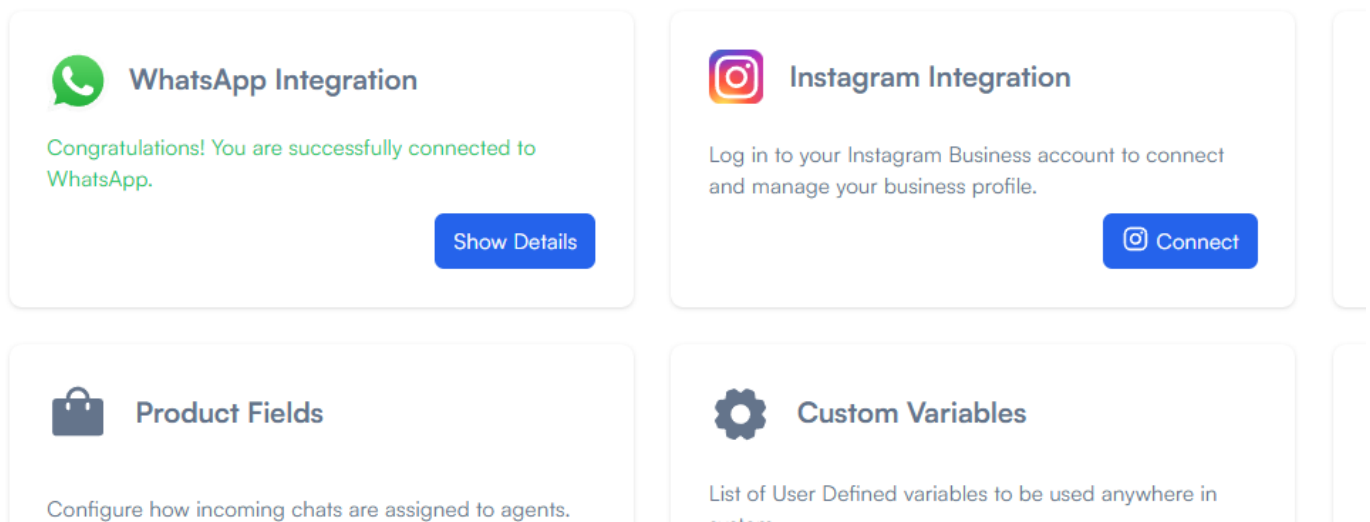
Instead of the manual method, the embedded signup method is the most convenient as it provides a step-by-step workflow to complete the whole process

To start, turn off the manual mode and click on Connect to get the embedded sign-up screen. this is a facebook.com link that will guide you through the whole process

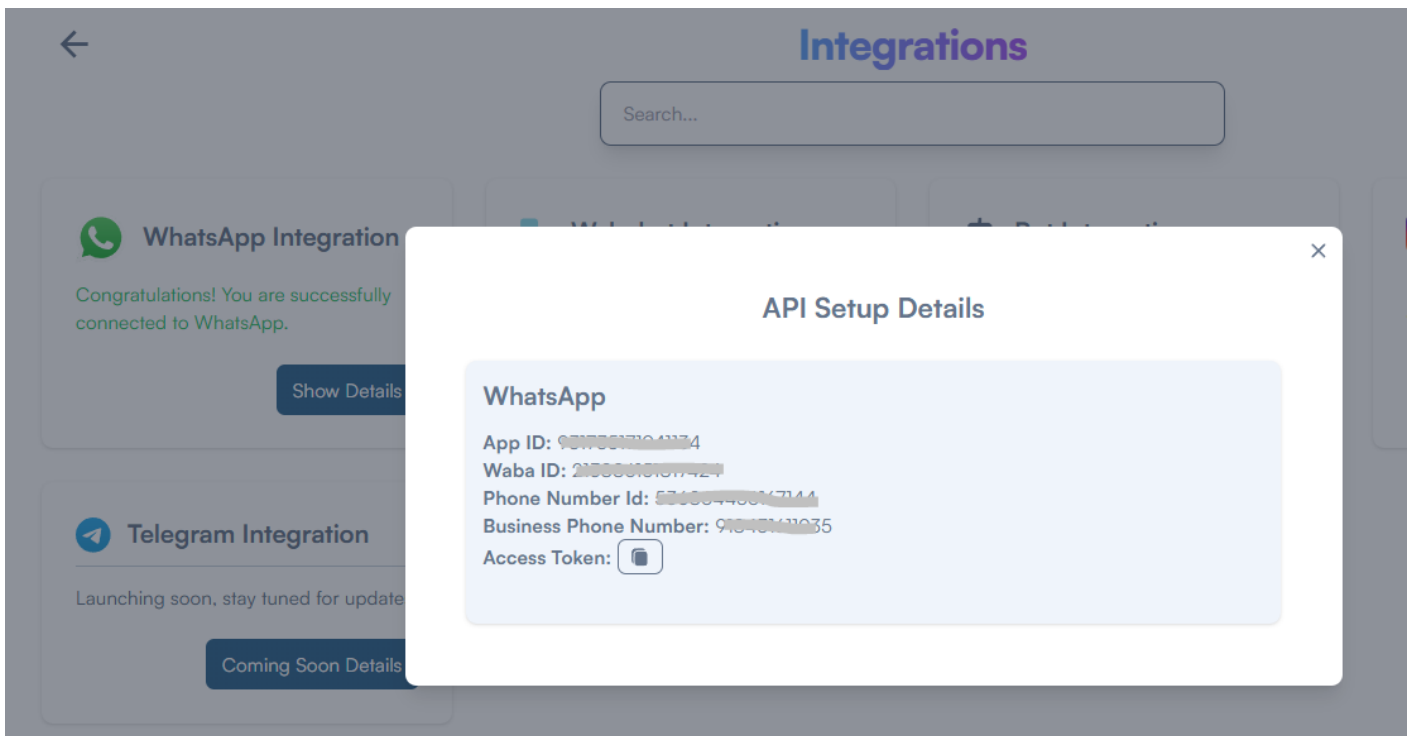


Once you have completed the process, your WhatsApp API will be generated and the integration will show complete

The Connect button will be displayed as "Show Details"



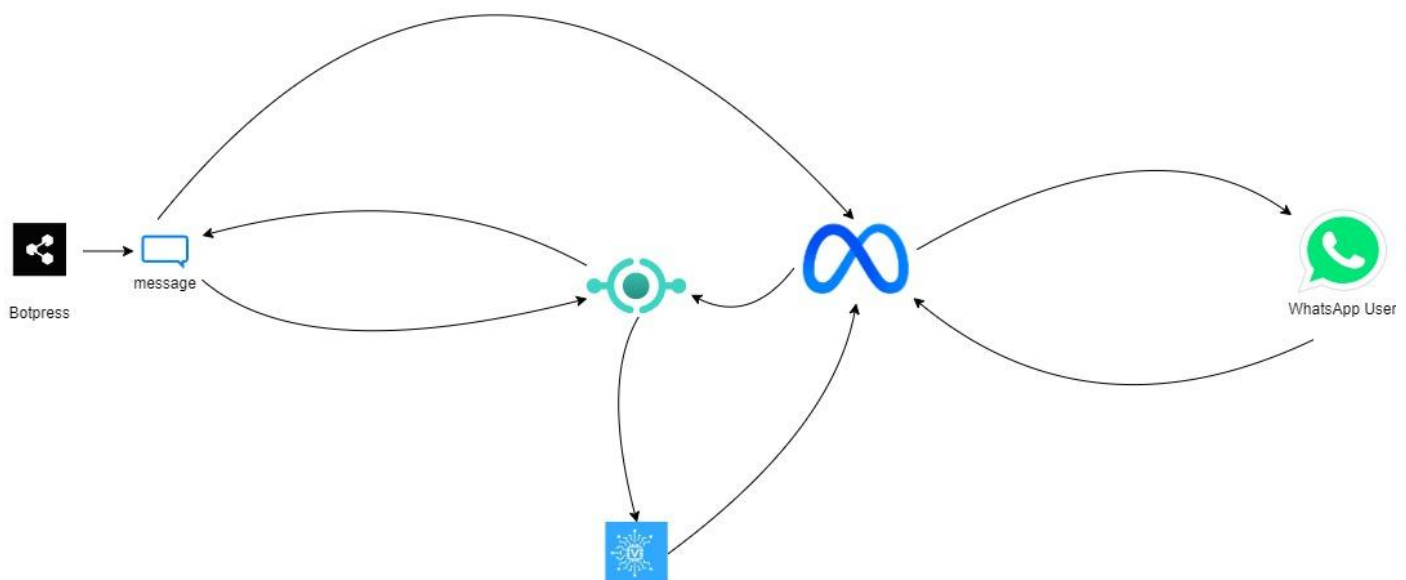
clicking on Show Details will reveal your WhatsApp credentials. copy the credentials to Botpress to complete your setup in Botpress



Now your Botpress can send messages via the bot to the user, but it is still not receiving messages from the user nor it is sending the bot response to vCX. To do so, follow the below steps

Connecting Botpress to vCX

The vCX Platform and its backend architecture is placed in between META Graph API's and Botpress



Select Bot Integration and select WhatsApp Integration

Step 1: Copy values from vCX WhatsApp Integration to Botpress WhatsApp Integration

Integrations

Search...



WhatsApp Integration

Congratulations! You are successfully connected to WhatsApp.

Show Details



Webchat Integration

Show Details



Bot Integration


Congratulations! You are successfully connected to Bot.

Show Details

The screenshot shows the Botpress Integrations page with a modal window titled "API Setup Details" open over the WhatsApp Integration card. The modal displays the following information:

- WhatsApp**
- App ID: 951700171041374
- Waba ID: 210000107017421
- Phone Number Id: 57400440047144
- Business Phone Number: 918401431035
- Access Token: [Copy icon]

From above, copy the Access Token and the Phone Number ID and paste them into the Botpress WhatsApp Integration, under Access Token & Phone Number ID respectively




WhatsApp

by botpress

2.0.6

Upgrade Integration




Something is missing or not working as expected with WhatsApp ?

Request changes

Configuration Info

Copy Share Link Uninstall

<https://webhook.botpress.cloud/2f32eaf9-ddbf-4085-aa9e-e42fb928a29f>



Authorize WhatsApp

Configuration

Use Manual Configuration

Verify Token

Token used for verification when subscribing to webhooks

b371dee120c2

Access Token

Access Token from a System Account that has permission to the Meta app

EAAFPrgm3QNBQ2qiCATMIFZBnGDZCviO2uYmxpkCXYDZCG7Kpa4wMbXdtGg24gQWi8srmFAJqLdGZCCZBUWn2ZBfgfg

Client Secret

Meta app secret used for webhook signature check

b371dee120c2

Phone Number ID

Default Phone used for starting conversations

393188703871783

Under Verify Token and Client Secret - add a random alphanumeric value as a token

Any random alphanumeric value can be used as the Verify Token & Client Secret

Saving the configuration will generate an error in Botpress, hence do not click **Save Configuration!**

393188703871783

Save Configuration

Step 2: Copy the Webhook and the Verify Token from Botpress to vCX

Configuration

Info

Copy Share Link

Uninstall

https://webhook.botpress.cloud/2000000-0000-0000-0000-e42fb928a29f

Authorize WhatsApp

Configuration

Use Manual Configuration

Verify Token

Token used for verification when subscribing to webhooks

b371dee120c2

Copy the webhook and the verify token from Botpress.

In vCX head to the Integration section and click Show Details for Bot Integration

←

Integrations

Search...

WhatsApp Integration

Congratulations! You are successfully connected to WhatsApp.

Show Details

Webchat Integration

Show Details

Bot Integration

Congratulations! You are successfully connected to Bot.

Show Details

Instagram

Manual

Log in to your Instagram Business account to connect and manage your business profile.

Connect

Paste the values copied from Botpress as shown

×

Select Integration

Currently integrated with Botpress

WhatsApp

Web Chat

Instagram

Telegram

Botpress WhatsApp webhook URL:

https://webhook.botpress.cloud/2f32eaf9-ddbf-4085-aa9e

✓

Botpress WhatsApp webhook secret

b37ldee120c2

Botpress webhook URL (Optional):

Botpress WhatsApp webhook secret

Botpress webhook secret

Enter botpress webhook secret

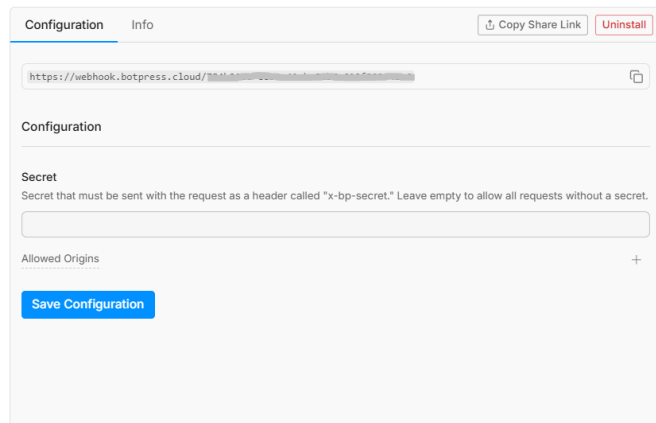
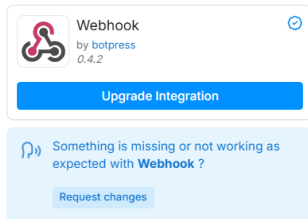
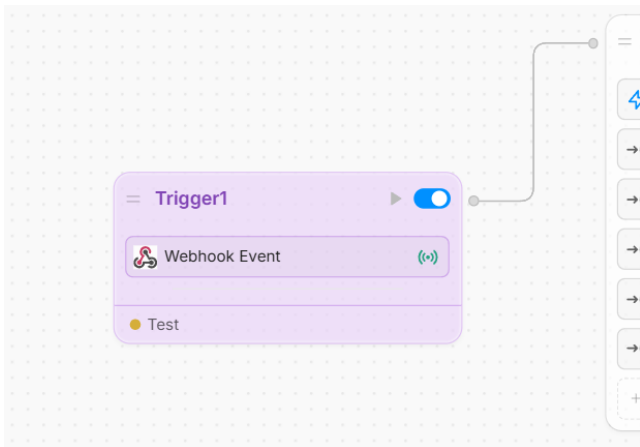
Add

Copy code

```
await axios.post(
  'https://webhook.versal.one/53d1bcd3-aaf0-45e1-9298-5395b6a27b61',
  {
    responsefrom: 'Bot',
    conversation: outgoingEvent,
    phonenumber: <users whatsapp number> # change this with the variable that contains users phone number
  },
  {
    headers: {
      'X-Access-Token': '5395b6a27b61'
    }
  }
)
```

Optional

If you have a trigger webhook event in Botpress, you need to add the Webhook URL in the Botpress Webhook URL section. This is optional



Save Configuration

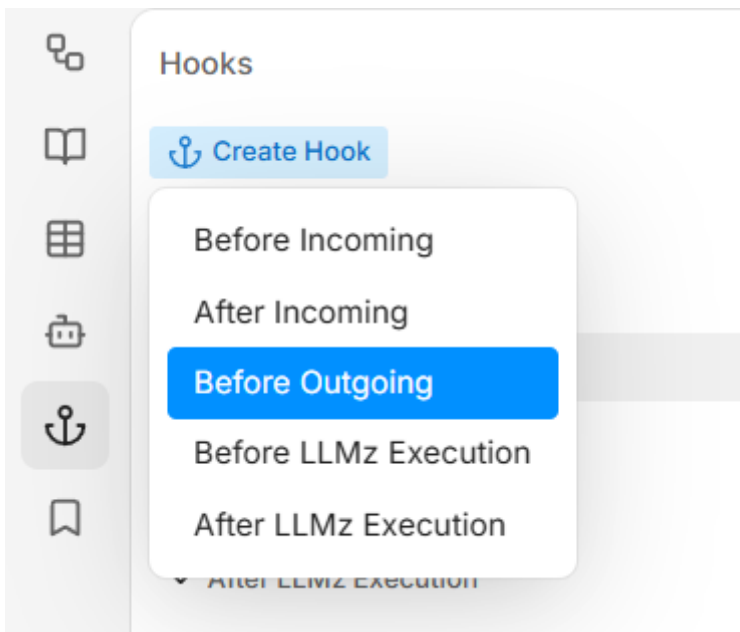
Click on the **Add** Button, you will receive a confirmation that the settings have been saved

Head back to Botpress and click on **Save Configuration**. The configuration will be saved successfully



Step 3: Create a Before Outgoing hook to receive chat messages in vCX

in Botpress under hooks add a new hook of type **Before Outgoing**



Give a name as shown

Create A New Hook

Name *

tovCX

Confirm

Hooks

- Before Incoming 0
- After Incoming 1
 - Tokens
- Before Outgoing 1
 - tovCX
- Before LLMz Execution 0
- After LLMz Execution 0

```
1 async function hook(outgoingEvent: IO.OutgoingEvent, event: IO.IncomingEvent): Promise<void> {
2   /** Your code starts below */
3
4
5
6   /** Your code ends here */
7 }
```

Start typing to get started! Here are a few ideas:

- // You can get/set the value of workflow variables ¹
workflow.SomeVariable = 'value'
- // Change values associated to a specific user ¹
user.ExampleEmail = 'email@website.com'
- // Use [Axios](#) to make calls to external APIs ¹
const { data } = await axios.get('https://api.com')
- // Use [Lodash](#) for data manipulation
const value = _.add(1, 2)
- // Use [Luxon](#) for all your date manipulations
const 1hAgo = luxon.DateTime.now().minus(60000)
- // Use the event object to access the user's message ¹
const userMessage = event.preview

Now head back to vCX and from the Bot Integration section, click "**copy code**" the code in the Integration panel and add it to your Botpress hook as shown below

```

1  async function hook(outgoingEvent: IO.OutgoingEvent, event: IO.IncomingEvent): Promise<void> {
2      /** Your code starts below */
3
4
5  await axios.post(
6      'https://webhook.versal.one/53d1bcd3-aaf0-45e1-9298-5395b6a27b61',
7      {
8          responsefrom: 'Bot',
9          conversation: outgoingEvent,
10         phonenumber: '5395b6a27b61',
11     },
12     {
13         headers: {
14             'X-Secret': '5395b6a27b61',
15             'Content-Type': 'application/json',
16             'User-Agent': 'Botpress'
17         }
18     }
19 );
20
21
22 /** Your code ends here */
23

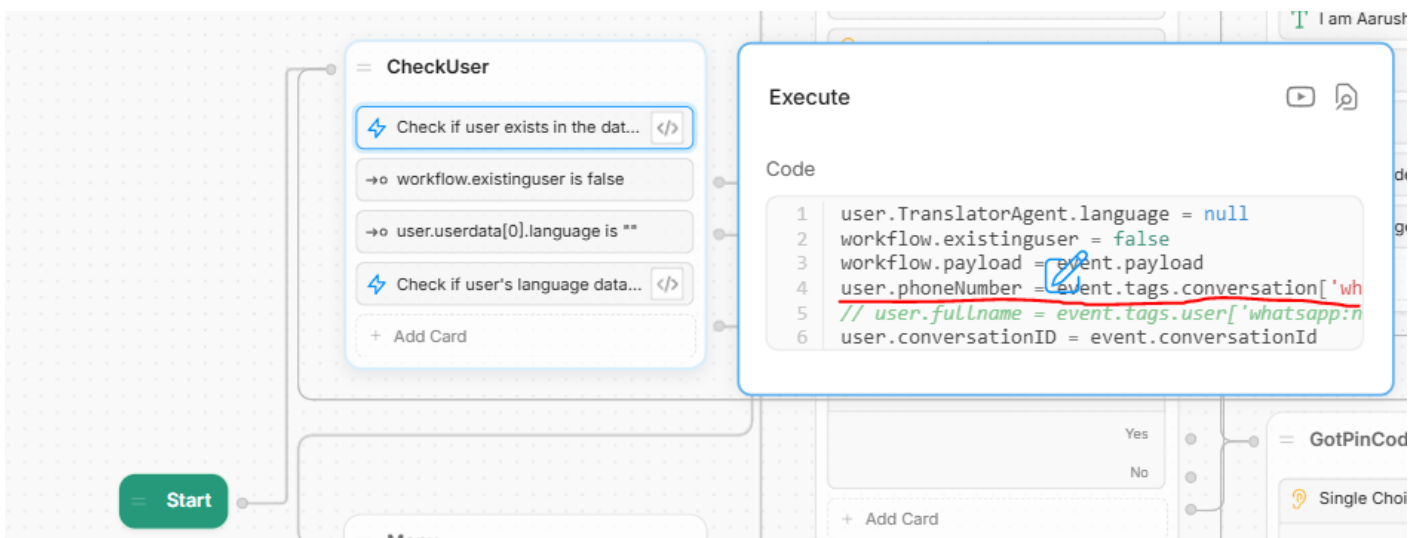
```

Delete the highlighted section shown above

Add your variable that holds the WhatsApp phone number of the user, chatting with the bot [in the below example, we are using user.phonenumber]

If you do not have a variable that stores the phone number of an incoming chat, you should add an execute code right after the Start Node

```
user.phonenumber = event.tags.conversation['whatsapp:userPhone']
```



```

1  ✓ async function hook(outgoingEvent: IO.OutgoingEvent, event: IO.IncomingEvent): Promise<void> {
2    /** Your code starts below */
3
4
5  ✓  await axios.post(
6      'https://webhook.versal.one/53d1bcd3-aaf0-45e1-9298-5395b6a27b61',
7  ✓  {
8      responsefrom: 'Bot',
9      conversation: outgoingEvent,
10     phonenumber: user.phonenumber
11  },
12  {
13  ✓  headers: {
14      'X-Secret': '5395b6a27b61',
15      'Content-Type': 'application/json',
16      'User-Agent': 'Botpress'
17  }
18  }
19  );
20
21
22  /** Your code ends here */
23  }

```

Important: you need to pass the WhatsApp Phone Number of the user conversing as a parameter, which without vCX platform will not be able to show the user details on the conversations tab.

In the above screenshot "user.phonenumber" is a variable that has the phone number of the user

Publish your bot to make the code changes work!

Now the setup is complete, to test the HITL feature you can easily toggle between Human and Bot in the conversation section

Total Conversations30

WhatsApp

HumanBot

Housing Baba

Phone: 9876543210

E-mail: housingbaba@gmail.com

Source: WhatsApp

Last Message: Dec 5, 2024, 09:27 PM

Search...

unknownBot

unknownBot

mpansari469Bot

Lijo Mathewadmin

SinghBot

12345678910Bot

Housing BabaBot

Tushar AgarwalBot

Dr. Ankit Krishnaniadmin

AstrologeBot

...

BotThu Dec 05 2024 14:11

Hello! I am vBot, your virtual assistant from Versalence. I'm here to guide you on how to streamline customer interactions and achieve seamless communication. How can I assist you today? Feel free to ask me about our solutions, pricing, or how to register for our free Starter Plan!

Contact UsThu Dec 05 2024 15:03

Hi Sir

Contact UsThu Dec 05 2024 15:03

Yes we are available

Contact UsThu Dec 05 2024 15:03

let me know how can we help you

Housing BabaThu Dec 05 2024 21:26

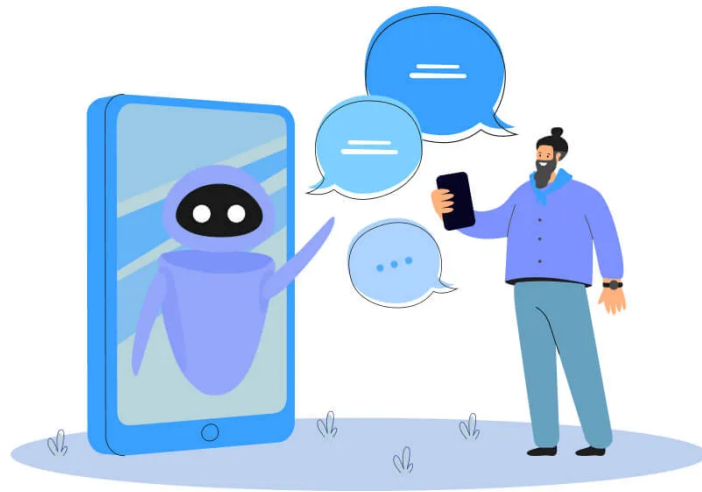
Hi

BotThu Dec 05 2024 21:26

Hello! I am vBot, your virtual assistant from Versalence. I'm here to guide you on how to streamline customer interactions and achieve seamless communication. Ask me about our solutions, pricing, or how to register for our free Starter Plan today!

Type something here

Custom Bot integration with vCX



Follow the steps to set up WhatsApp API using the manual method or embedded signup

A webhook to receive incoming messages

Incoming Message format

Below is the format of incoming messages payload as received from META webhook and will be forwarded to your Bot webhook

```
{
  "object": "whatsapp_business_account",
  "entry": [
    {
      "id": "xxxxxxxxxxx",
      "changes": [
        {
          "value": {
            "messaging_product": "whatsapp",
            "metadata": {
```



```

    "display_phone_number": "xxxxxxxxxx",
    "phone_number_id": "xxxxxxxxxx"
  },
  "contacts": [
    {
      "profile": {
        "name": "Sanjeev Reddy"
      },
      "wa_id": "xxxxxxxxxx"
    }
  ],
  "messages": [
    {
      "from": "xxxxxxxxxx",
      "id": "wamid.HBgMOTe5NzQxMzkxNzAwFQIAEhgUM0Y2OTU4Qzc0OEYyMkMzQ0NCOEIA",
      "timestamp": "1733564481",
      "text": {
        "body": "hi"
      },
      "type": "text"
    }
  ]
},
"field": "messages"
}
]
}
]
}

```

Outgoing Messages to the user

format to send data to META using vCX

```

curl --location -X POST 'https://api.versal.one/<client-id>' \
--header 'Authorization: Bearer <client-token>' \
--header 'Content-Type: application/json' \
--data '{
  "purpose": "sendmessage",
  "message_data": {
    "messaging_product": "whatsapp",

```

```

    "recipient_type": "individual",
    "to": <recepient>,
    "type": "text",
    "text": {
      "body": <Message or response from Bot>
    }
  }
}'

```

Outgoing messages to vCX

format to send data to vCX as a bot response

```

curl --location -X POST 'https://webhook.versal.one/<client-id>' \
--header 'Authorization: Bearer <client-token>' \
--header 'Content-Type: application/json' \
--header 'User-Agent: 'Bot' \
--data '{
  "responsefrom": "Bot",
  "conversation": {
    "id": "0470d58f-ad45-4d16-b229-76476e598fed",
    "type": "text",
    "channel": "channel",
    "direction": "outgoing",
    "payload": {
      "type": "text",
      "text": "Hello! I am Bot, your virtual assistant",
      "markdown": true
    },
    "preview": "Hello! I am Bot, your virtual assistant",
    "incomingEventId": "48621030-b469-420e-b827-9b9b293ba312"
  },
  "phonenumber": "xxxxxxxxxx"
}'

```