

INDEX

Introduction	2
- About the Author	
- What is ChatBot	
- Why ChatBot	
- How ChatBot is working	
- Benefits of ChatBot	
- Future of ChatBot	
- ChatBot Platforms in Market	
- Types of ChatBot	
ChatBot Framework (Twilio Autopilot)	4
- About Twilio	
- Why AutoPilot	
- Autopilot Framework in Detail	
- Tasks	
- Tasks Default Behaviours	
- Field Types	
- Train Task	
- Actions	
- StyleSheet	
- Simulator	
- Queries	
- Channels	
- Webhooks	
- Settings	
Building Conversational ChatBot	15
- Start Building Conversational Bot	
Building DataCollecting ChatBot	24
- Start Building Data Collecting Bot	
Facebook Channel Integration	30
- Pre-requisites	
- Process of Integration	
Business Ideas Around ChatBots	33
- List of Ideas	
Demo Videos of Real Projects	35
- Video Links	
- Audio Links	

Introduction

About Author

Hi, I am Ankit Jayswal. Having 12+ Years of experience in various Domains like **VoIP | AI | ML | ChatBots Computer Vision**. Core experience in Design, Develop and Deploying solutions in these technologies. My core experience is in VoIP Technology, so you might have a question about what things drag me towards AI also. So, the answer is I worked with many VoIP providers. Out of them, one of my favourites is **Twilio**. I am not a promoter of any platform here but I am just sharing my experience and journey towards AI. How exactly I moved towards this AI ChatBot technology and then towards Machine Learning prediction models and computer vision. So, gradually I came across the AutoPilot product of Twilio, as it is getting integrated with Voice channel I got interested in exploring it more. Then I created some applications of Voice IVRs for various Domains. These Voice IVRs we can say VoiceBot also work on voice commands, which can replace traditional press1 press2 IVR what we have right now in the market. Then I was inspired by the Python language as I need to explore more about Artificial Intelligence. Likewise I explored more about Data Science and Machine Learning Algorithms and predictions models. Okay, so I think this is enough information I have given about me, to relate with me while reading this book. Later in this book we will purely discuss ChatBots (based on AutoPilot Platform) Development and their practical use cases.

I am writing this content to share my Realtime experience of ChatBots Development. **And also will share some Business Ideas around it.**

I believe that If the knowledge can not be useful to real-life problems then it is not of much worth.

Those business ideas you can read and share your valuable feedback for that with me. I have given my reachability details below.

If you like any Business Idea and want to implement for yourself or someone else I will help you completely with proper source code deployment and get it all ready for you. Also if you are an Entrepreneur and interested to do business with that we can work together.

Contact Details:

Email: ankitjayswal87@gmail.com

Mobile: +91 9979272423

WhatsAPP: +91 9979272423

Professional Details:

Linked In: <https://www.linkedin.com/in/ankitjayswal/>

Upwork Profile:

<https://www.upwork.com/freelancers/~01ced4cf7cd6c769e5?s=1044578476142100518>

GitHub: <https://github.com/Data-Science-Machine-Learning>

What is ChatBot

Chatbots are Virtual Assistants that can work 24/7 for us and Answer customer queries like a Human. It is basically Artificial Intelligence which is able to understand the intent of the user query and able to Answer like a human. We can train this Intelligence based on Data we received every time by the user and make it more smarter over a time. It can be integrated with Communication Channels like Voice, SMS, WhatsApp, Facebook etc. Implementing and Utilising ChatBots capabilities Sky is the limit.

Why ChatBot

Nowadays Chatbots are very essential for every Website to handle routine customer queries without human intervention. Chatbots can be trained as per our need and can be made capable to answer all customer questions like product service information, costing and pricing plans etc. Chatbots can help All businesses to provide quick customer support and satisfaction. There are many more benefits of chatbots and reasons to use it. That you gradually understand, as you go further in reading the book ahead.

How ChatBot is working

Chatbot is like an Artificial Human which accepts text or voice from the user. If it is a chatbot then it accepts text and if it is a voicebot then it accepts voice data. If it is a voicebot then text to speech and speech to text also come in the picture. It processes the natural language and predicts the user's intent. And based on intent classification the chatbot responds with specified replies. This is the basic working pattern of a chatbot.

- Accept data (text or voice)
- Text to speech, Speech to text (If VoiceBot)
- Natural Language Processing
- Natural Language Understanding
- Predicts Intent of user based on input text
- Route conversation to detected Intent
- Reply with predefined responses for the Intent

Benefits of ChatBot

- Instant Customer Support
- No Waiting Time for Customer
- 24/7 Available
- Reducing Agent's Workload (No need to Answer every customer immediately)
- Filtered Traffic reaches to Actual Agent

Future of ChatBot

If we get a chance to use such AI ChatBots on our favourite communication channels like Facebook, WhatsApp then people will love it. Instead of going to different websites and installing new applications on devices, people will love to use ChatBot services on their favourite communication channels.

As people don't have to Download here extra applications on their devices, they will also get an interactive way of talking to Intelligence. So, in the future there is a huge opportunity to work around ChatBots. Few examples are like Appointment Booking, Order Booking use cases. Assume you can book your Appointment just by chatting with a Virtual Agent that can

work for you 24/7 and guide you for your basic queries also. Assume you can book your Bus or Railway tickets over WhatsApp channel just by chatting.

ChatBot Platforms in Market

There are a lot of ChatBot frameworks available to work with from our Tech Giants like Amazon Lex, Google Dialog Flow, Twilio Autopilot and many more. These are all good and widely used frameworks nowadays.

Later in this book I will share my experience with the Twilio Autopilot framework which I used in many real time live projects for the Real requirement.

Types of ChatBot

Below are two main categories we can define for ChatBots types. They are as below.

a) Conversational / Informative Chatbots

This is a very simple kind of ChatBot. It understands the user's intent and returns an appropriate reply for that. For example, someone says "**I want to know about your services**", "**Let me know about your services**", "**What services your company provides**" for all such occurrences Bot will give the response associated with the particular intent called "**our-services**", like "**We provide various AI Services for all Verticals, and these are some use cases of our services.**" These kinds of chatbots are very useful to resolve routine customer queries and FAQs.

b) Order taking Chatbots

These can be complex chatbots examples. Some examples are **Pizza Ordering, Coffee Ordering**, Any kind of **Appointment Booking** use case. Here you can build small online orders taking eCommerce stores also for your brand. Like Grocery stores, Vegetables & Fruits stores.

ChatBot Framework (Twilio Autopilot)

About Twilio

Twilio is a very Robust cloud based Telephony service provider. It is much more than the only telephony service provider. A lot of voice, sms solutions and integrations are available in that. One of solutions called **AutoPilot** is the conversational AI platform provided by Twilio. That is provided with many communications channels like Voice, SMS, WebApp, MobileApp, Slack, Facebook, WhatsApp. Means once you make your Bot with that you can connect it with these many communications channels easily.

Why AutoPilot

In the market many platforms are available but personally I like this because it has very Good structured architecture and is easily understandable. It has many customization opportunities available. Supports many programming languages in custom Development. Like PHP, Python, nodejs etc. **Mainly I need to produce valid JSON action to work with Autopilot.**

Autopilot Framework in Detail

It is very easy to build ChatBots around Autopilot. It works on HTTP requests only. We just need to produce a proper JSON response for HTTP requests. These JSON responses are the actions of a chatbot that defines what to do next within a conversation. Below I will share all the valuable actions needed to build basic ChatBot to more advanced ChatBot.

Below are some key components of the framework and have given Definitions along with examples. I tried to explain it in as much simple language as possible. So, let's have a look at these components. **If you are not able to understand these components 100 percent immediately, just do not worry, later all will be explained with the examples and screenshots. Obviously Practical is more understandable then Theory.** Just keep reading and have some patience to learn.

Tasks

Here Tasks you can assume like the user's intent. We can create various tasks like “welcome”, “goodbye”, “book_order”, “change_address”, “know_about_services”. The tasks are getting invoked or triggered by the user's input during chat. Like, if the user says “**Please book my order**” then the “**book_order**” task will be triggered and the task will have a particular Action to be executed. Below is the image for sample tasks of Bot.

The screenshot shows the Autopilot console interface. The top navigation bar includes 'Console', 'ankit_jaywal87@yahoo.com's Account', a search bar, and links for 'Account', 'Billing', and 'Ankit'. The left sidebar has tabs for 'Develop' and 'Monitor', with 'Develop' selected. Under 'Develop', there are options for 'Add a task', 'Find a task', 'No pinned products yet!', 'Explore Products', 'Autopilot (US1)', and 'CustomerCareBot'. The 'Tasks' section is highlighted in the sidebar. The main content area displays a table of tasks for the 'CustomerCareBot'.

TASKS	SAMPLES	FIELDS			
appointment_booking	1	0	Program	Train	
timings	2	0	Program	Train	
services	2	0	Program	Train	
goodbye	17	0	Program	Train	
fallback	0	0	Program	Train	
greeting	16	0	Program	Train	
collect_fallback	0	0	Program	Train	

At the bottom of the console, there are buttons for 'Build model' and 'Simulate', along with a status indicator 'Model is up to date'.

Tasks Default Behaviours

There are 3 default behaviours it provides. Basically these are the conditions on which a particular default task can be executed. Below are the default behaviours and it's description.

Bot Initiation:

This is the starting point of your Bot. For VoiceBot this will be the starting point. This can be like a greetings task or welcome task.

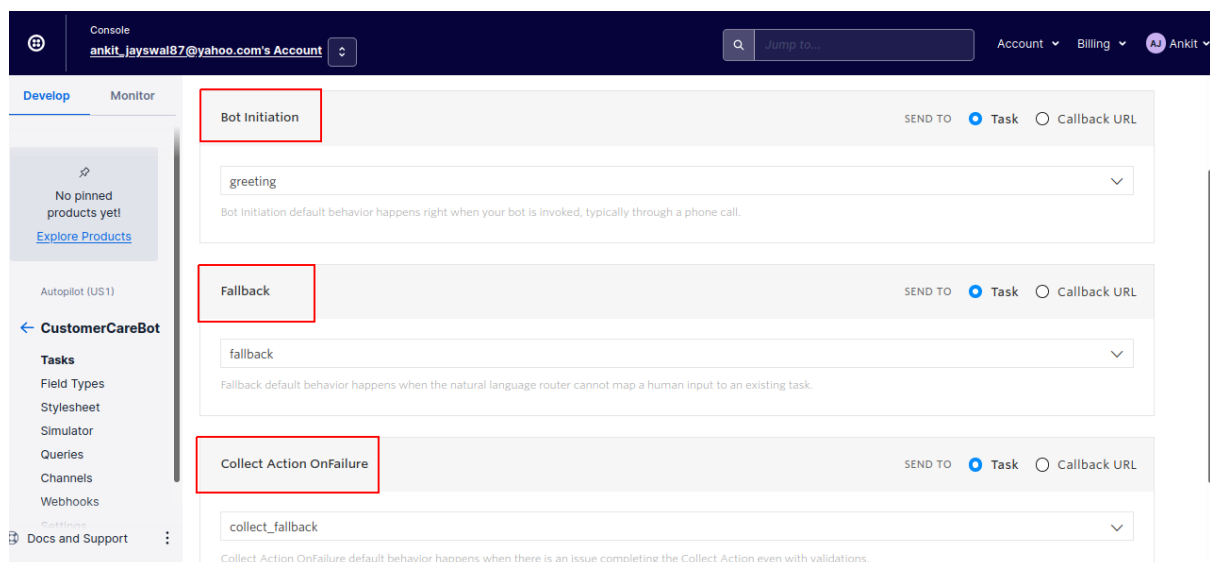
Fallback:

This is invoked when Bot can not understand your natural language. You can select some default task when Bot is not able to understand the user input phrase. Here you can set user guiding tasks like what they can ask to Bot. E.g. I did not understand it properly. You can ask us about our services, features and book order.

Collect Action OnFailure:

Whenever the collect action fails to collect inputs this can be the task to be executed.

Just do not worry, later all things will be explained from scratch while Building ChatBots.



Field Types

By default it provides some builtin data types. These are as follows:

YES_NO
NUMBER
DATE
TIME
FIRST_NAME
LAST_NAME
EMAIL
MONTH
DAY_OF_WEEK
STATE
COUNTRY

CITY
CURRENCY
LANGUAGE
NUMBER_SEQUENCE
ALPHANUMERIC etc.

Here you are also able to create your own Field Type to be recognized in tasks like **collect**. For example we can create a Custom Field Type called **CarType** and I can add values to it like Maruti, Tata, BMW.

So here CarType is the custom field and Maruti, Tata, BMW are the values of it. We can use this data type in **collect action** and we are able to validate user input with three given values, the value other than these three will not be accepted by Bot. Just like we are providing form validation in web form. You can visit this link to get more information about Field Types, <https://www.twilio.com/docs/autopilot/built-in-field-types>

Train Task

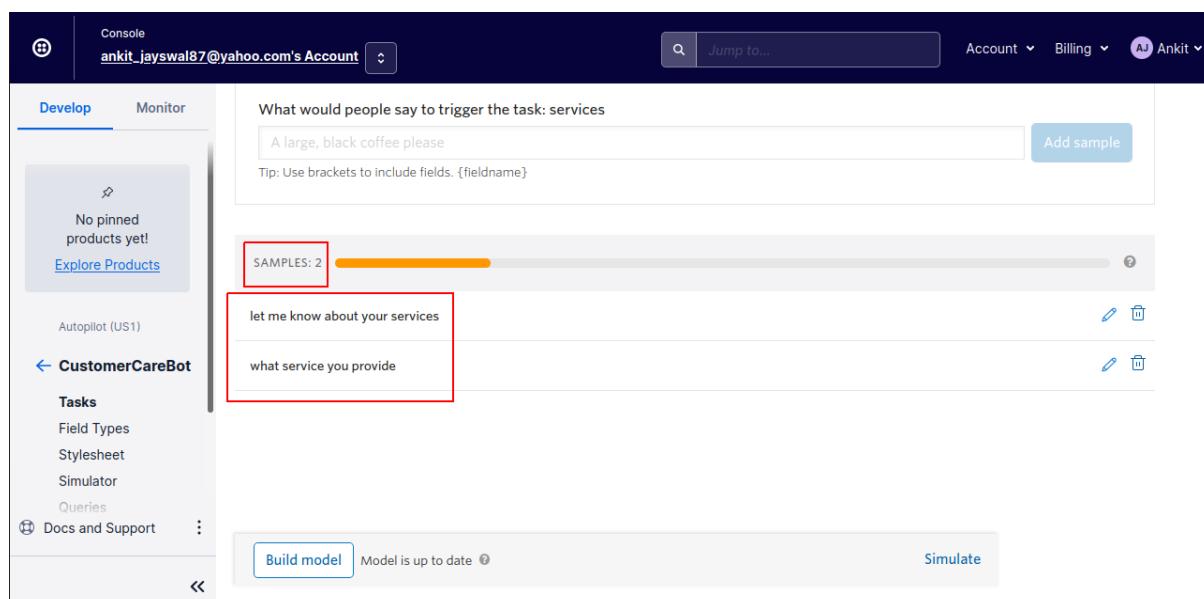
As said above, the task is invoked by some user inputs and these inputs are called sample data. We need to add more and more sample data to train our task. So our task is getting invoked properly on user input.

For example above we can add below sample data for the task to be trained,

- **book my order**
- **please book my order**
- **can i book my order**

These phrases (data) are called sample data which is used to train ChatBot to make it smarter. These data we can add for the tasks defined.

In the image below I have added some sample data for the **services** task, just have a look at this. Remember **services** task, so the sample data are different in image then above mentioned data.



Actions

Every task has some Actions associated with it. These actions are the real execution of the tasks. Twilio Autopilot provides various actions to be executed, once the task is triggered. So, some of the actions are 'say', 'redirect', 'collect', 'play', 'listen', 'handoff', 'remember', 'show'. **These are the Building blocks of your ChatBot. Need to pay more attention here.** Need to use these parts smartly to build our chatbots effectively. Below in the image I have added Say action, just have a look at that.

The screenshot shows the Twilio Autopilot console interface. At the top, there's a header with the account name 'ankit_jayswal87@yahoo.com's Account' and a search bar. Below the header, there's a navigation bar with 'Develop' and 'Monitor' tabs. The main content area is titled 'All Tasks > services'. It contains a form for configuring a task. The 'UNIQUE NAME' field is filled with 'services' and the 'FRIENDLY NAME' field is filled with 'Optional'. There's a 'Switch to train task' button. Below this, there's a section for 'Program what the task does when triggered.' with a toggle for 'USE' set to 'ActionBin'. The 'actions' field is a JSON array containing a single 'say' action with the text 'We are IT services and making solutions to help small businesses at affordable prices.' At the bottom, there are buttons for 'Save', 'Build model', and 'Simulate'.

Say Action:

This action will simply do text to speech if it is a VoiceBot and reply with a text message if it is a ChatBot. The only say action will terminate the session once executed. So this action can be used in end statements like say for example "Thank you for your Order. Your Order number is 238438" Session means every new conversation starts with some unique reference id.

```
{
  "actions": [
    {
      "say": "Hello World!"
    }
  ]
}
```

Say Action with listen:

This will do the same as say action but will not terminate the session. It will say and then wait for user input. Once a user says something it will try to find another task to follow based on user input.

```
{
  "actions": [
    {
      "say": "Hello World!"
    }
  ]
}
```



```

    },
    {
        "listen": true
    }
]
}

```

Play Action:

This task plays the pre-recorded file to the caller. Works with VoiceBot.

```

{
  "actions": [
    {
      "play": {
        "loop": 2,
        "url": "https://api.twilio.com/cowbell.mp3"
      }
    }
  ]
}

```

Collect Action:

This is like a web form we fill on a website. You can assume it as a data collector that accepts different types of data. It provides some builtin data types like Number, Date, Time, FirstName, LastName, Phone number etc. So with these data types you can validate user data and accept it. For example we need to get details of visitors like Name, Number and Address then we can create one collect action which will ask all three questions and once completed it will redirect to some url to retrieve collected data and store and process.

```

{
  "actions": [
    {
      "collect": {
        "name": "collect_comments",
        "questions": [
          {
            "question": "What is your full Name ?",
            "name": "full_name"
          }
        ],
        "on_complete": {
          "redirect": "https://mywebserver.com/collect.php"
        }
      }
    }
  ]
}

```

Handoff Action:

This will work with VoiceBot only. Sometimes we need to transfer a call to a Live Agent if a customer wants to talk with a Live person. In this case we can use this action. You can call this action to finally close the Deal with the customer. Here we need to provide one URL which can produce XML output to connect (Dial) real person number.

```

{

```

```

"actions": [
  {
    "say": "Hold on, we are connecting you with an agent"
  },
  {
    "handoff": {
      "channel": "voice",
      "uri": "INSERT YOUR TWIML_URL HERE",
      "method": "POST"
    }
  }
]
}

```

Redirect Action:

This we can use to customize our response more effectively. Suppose our ChatBot works with both **Voice** and **Facebook** channels. Then our welcome message can be different for both cases like

“Welcome to ABC company, please **say** how can I help you?” for the Voice Bot and

“Welcome to ABC company, please **type here** how can I help you?”

Or another example like If you lookup customer details and if an existing customer then greet with his/her name versus greet with a common message. For known customers, messages could be

“Dear, **Ankit Jayswal**. How can I help you?” and for new customer messages could be

“Dear **customer**, How can I help you?”

```

{
  "actions": [
    {
      "redirect": {
        "uri": "https://mywebserver.com/welcome.php",
        "method": "POST"
      }
    }
  ]
}

```

Remember Action:

This is for storing key-value information during the conversation. It remains stored during conversation until rewrite. This value you can use later in conversation if required.

```

{
  "actions": [
    {
      "remember": {
        "first_name": "ankit"
      }
    }
  ]
}

```

Show Action:

This action is available for providing Rich replies like you can respond with an image with proper body and message and label to it. It works with supported chatbot channels only like web, facebook.

```
{
  "actions": [
    {
      "show": {
        "body": "Twilio Owls",
        "images": [ {
          "label": "Original Owl",
          "url": "https://demo.twilio.com/owl.png"
        } ]
      }
    }
  ]
}
```

StyleSheet

This setting provides the below json setting. Here you can set male female voices from Polly service. Right now below **Polly.Matthew** male voice is set. Some available female voices are Polly.Salli, Polly.Joana, Polly.Kendra Polly.Kimberly and male voices are Polly.Matthew, Polly.Joey etc. Also when we use collect action in conversation we can allow a number of attempts and retry message text. This is a general message setting invoked for every collect action if not defined explicitly.

```
{
  "style_sheet": {
    "voice": {
      "say_voice": "Polly.Matthew"
    },
    "name": "",
    "collect": {
      "validate": {
        "on_failure": {
          "messages": [
            {
              "say": {
                "speech": "I didn't get that. What did you say?"
              }
            },
            {
              "say": {
                "speech": "I still didn't catch that. Please repeat."
              }
            },
            {
              "say": {
                "speech": "Let's try one last time. Say it again please."
              }
            }
          ]
        },
        "repeat_question": false
      },
      "on_success": {
        "say": {

```

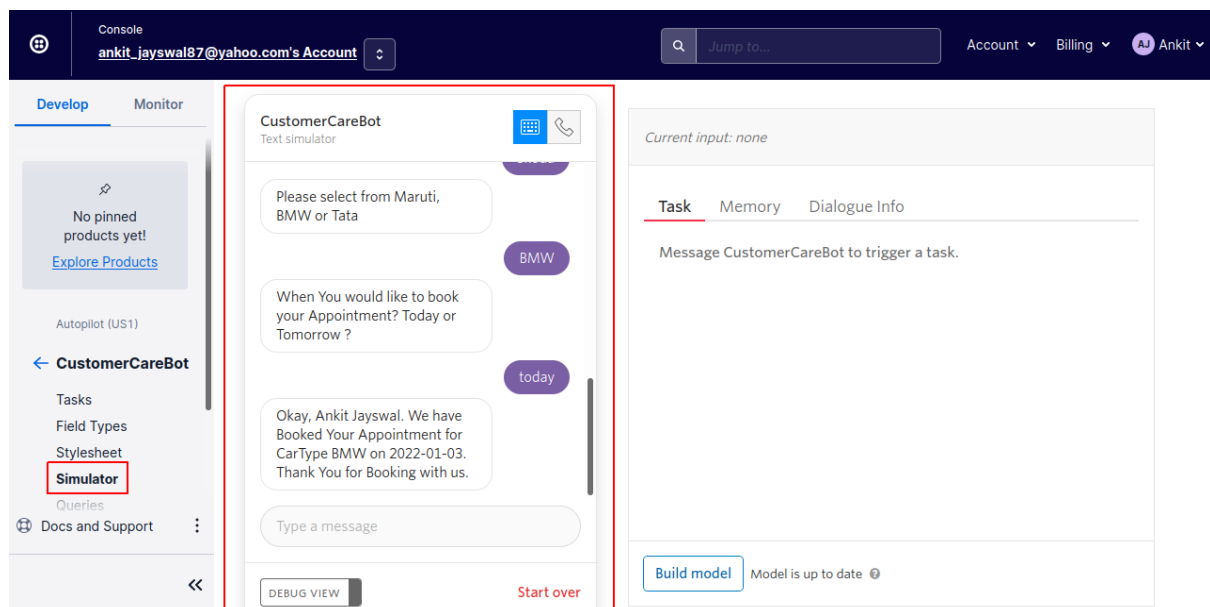
```

    "speech": ""
  },
  "max_attempts": 4
}
}
}
}
}

```

Simulator

This is for the testing of Bot. Once you develop ChatBot you can test it all on this simulator before deploying it to production. This will provide you a better testing path and you do not want to play with original FB, Voice or SMS kind of channels. This will help you in eliminating real cost while testing the Bot before making it live. Below is a sample image for simulator.



Queries

Here you will find user input logs and these user input phrases you can assign to particular appropriate tasks also if you wish. This will help you to make your Bot more smarter gradually. You can see the confidence level of tasks executed for the user input. In the image below you can see a list of queries.

Console

ankit_jayswal87@yahoo.com's Account

Jump to...

Account

Billing

AJ Ankit

Develop

Monitor

products you

Explore Products

Autopilot (US1)

CustomerCareBot

Tasks

Field Types

Stylesheet

Simulator

Queries

Channels

Webhooks

Settings

Docs and Support

STATUS

VALUES

TASKS

CHANNELS

All Status

All Tasks

All Channels

Actions

50 Per Page

QUERY	VALUES	TASK	CONFIDENCE	DATE	STATUS
today Chat	2022-01-03 Add	no task Assign		11:54am IST Jan. 03, 2022	Pending
BMW Chat	BMW Add	no task Assign		11:54am IST Jan. 03, 2022	Pending
Skoda Chat	none Add	no task Assign		11:54am IST Jan. 03, 2022	Pending
9979271234 Chat	9979271234 Add	no task Assign		11:54am IST Jan. 03, 2022	Pending
Jayswal Chat	Jayswal Add	no task Assign		11:54am IST Jan. 03, 2022	Pending

Build model

Model is up to date

Simulate

Channels

These are the channels with which you can integrate your Bot. There are different methods of integration available for different channels. We will discuss one Facebook channel integration in this book later. It supports the below channels right now.

Voice

SMS

Web Chat

Alexa

Google Assistant

WhatsApp

Facebook Messenger

Console

ankit_jayswal87@yahoo.com's Account

Jump to...

Account

Billing

AJ Ankit

Develop

Monitor

products you

Explore Products

Autopilot (US1)

CustomerCareBot

Tasks

Field Types

Stylesheet

Simulator

Queries

Channels

Webhooks

Settings

Docs and Support

Programmable Chat

Alexa

Google Assistant

WhatsApp

Facebook Messenger

Webhooks

Here you can monitor some Bot events in realtime. You can select a list of events and provide your web URL to receive these events on it. It provides below events to monitor in real time. This is all for monitoring purposes, how it works in real time.

onDialogueStart

onDialogueEnd

onDialogueTaskStart

onDialogueTaskEnd

onDialogueTurn

onCollectAttempt

onActionsFetch

The screenshot shows the 'Webhooks' configuration page for a bot named 'CustomerCareBot'. The left sidebar has 'Webhooks' highlighted. The main area contains a form with the following fields: 'UNIQUE NAME' (containing 'analytics-webhook'), 'CALLBACK URL' (containing 'https://example.com'), and a 'POST' dropdown. Below these are checkboxes for various events: onDialogueStart, onDialogueEnd, onDialogueTaskStart, onDialogueTaskEnd, onDialogueTurn, onCollectAttempt, and onActionsFetch. A 'Create' button is at the bottom left of the form area.

Settings

Here it provides options to rename your Bot, add description, enable/disable log queries and delete Bot. You can see the same things in the image below.

The screenshot shows the 'Settings' page for the 'CustomerCareBot'. The left sidebar has 'Settings' highlighted. The main area contains the following sections: 'Bot Identity' with fields for 'UNIQUE NAME' (containing 'CustomerCareBot') and 'DESCRIPTION OR FRIENDLY NAME' (containing 'CustomerCareBot'); 'Opt-ins' with a checked 'Log queries' checkbox; and 'Deleting your bot' with a 'Delete this bot' button. A 'Save' button is located below the 'Opt-ins' section.

Building Conversational ChatBot

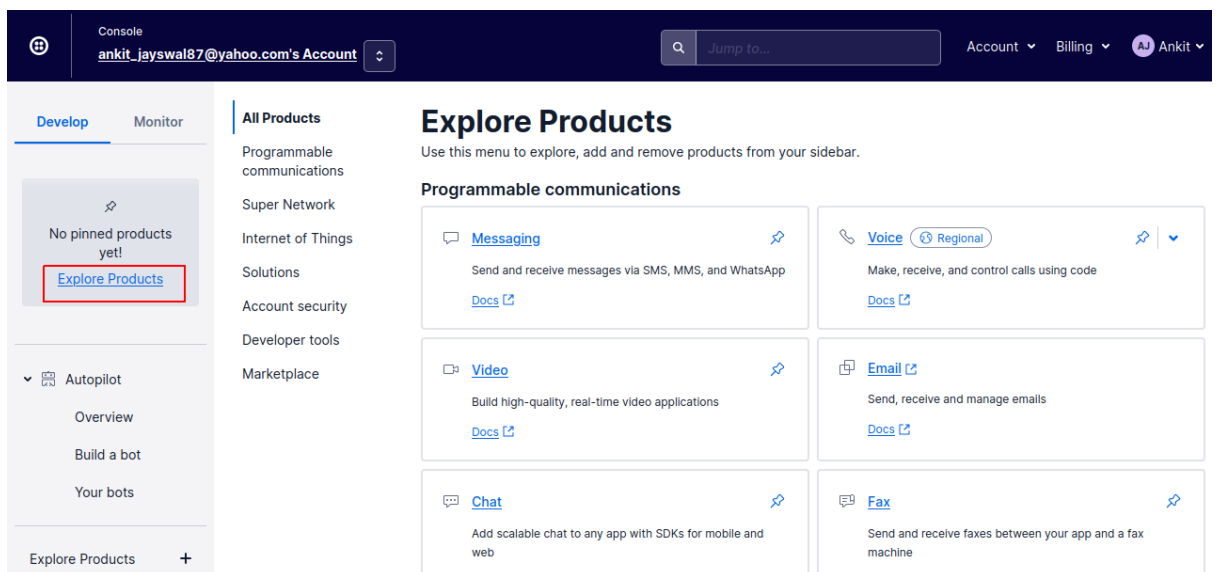
So far we understand various components of chatbots. Now it is time to start building chatbots with the Twilio Autopilot platform. You will require live Twilio Account Login for this. To signup new Twilio Account you can go to this URL <https://www.twilio.com/try-twilio> Once you have a working Twilio Account you can start making chatbots.

Start Building Conversational Bot

Let's make one simple conversational Bot like helping in replying to customer routine queries. We will try to cover questions like, **“What services I can avail from you”, “what are the timings”**, etc. Let's call this Bot as Customer Care Bot. So below is the process we will follow to make such a basic Bot. I assume here you have a working Twilio Account.

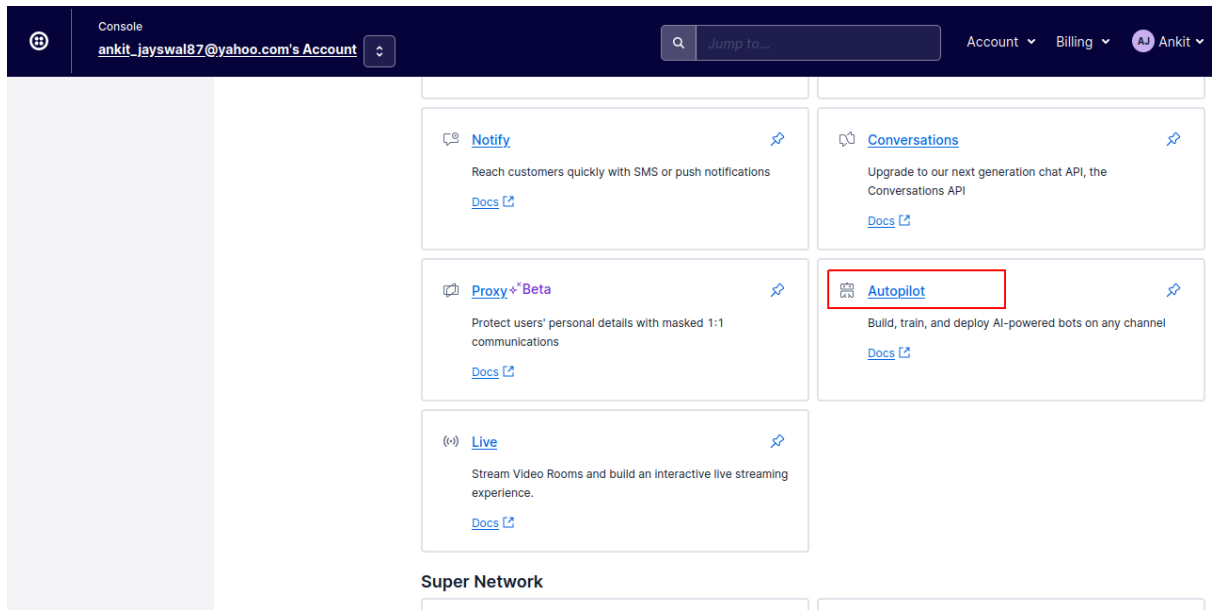
1) Login to Twilio Account

Once you Login to your Twilio Account you will see a screen like Below. Click on Explore Products as shown in the image below.

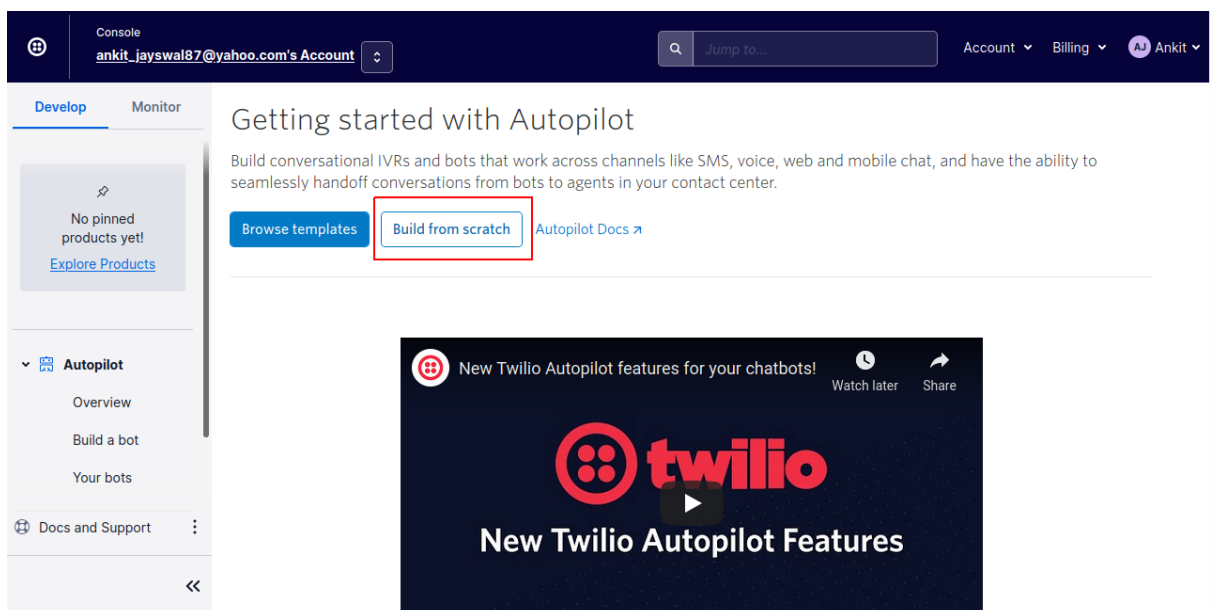


2) Goto Autopilot

Now you need to click on Autopilot as shown below.

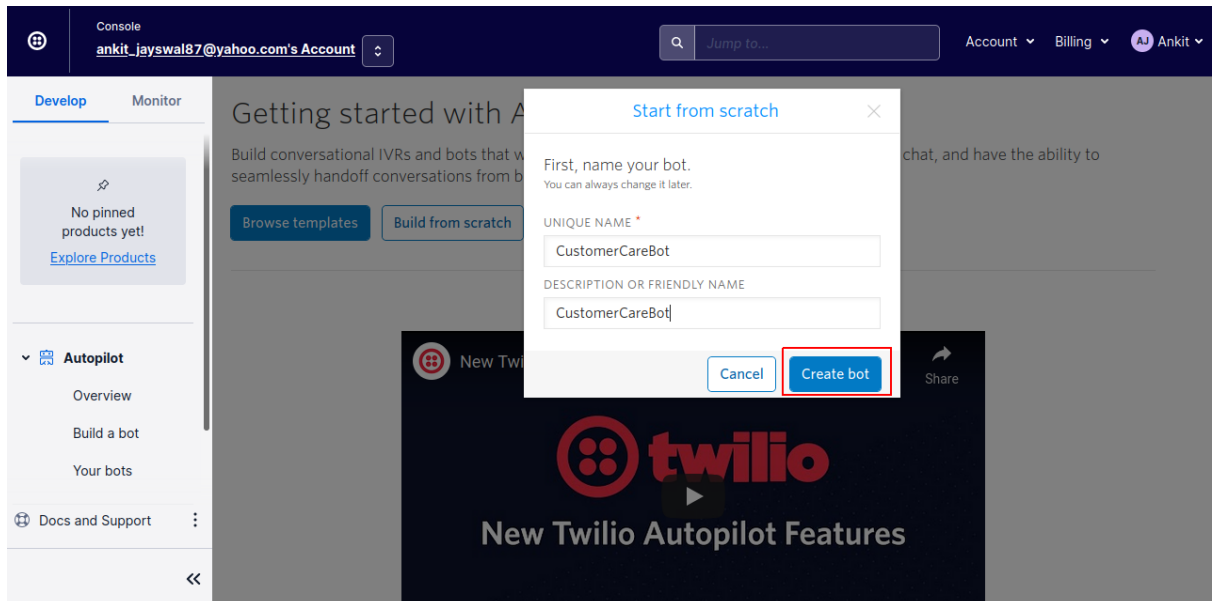


3) Click on Build from scratch

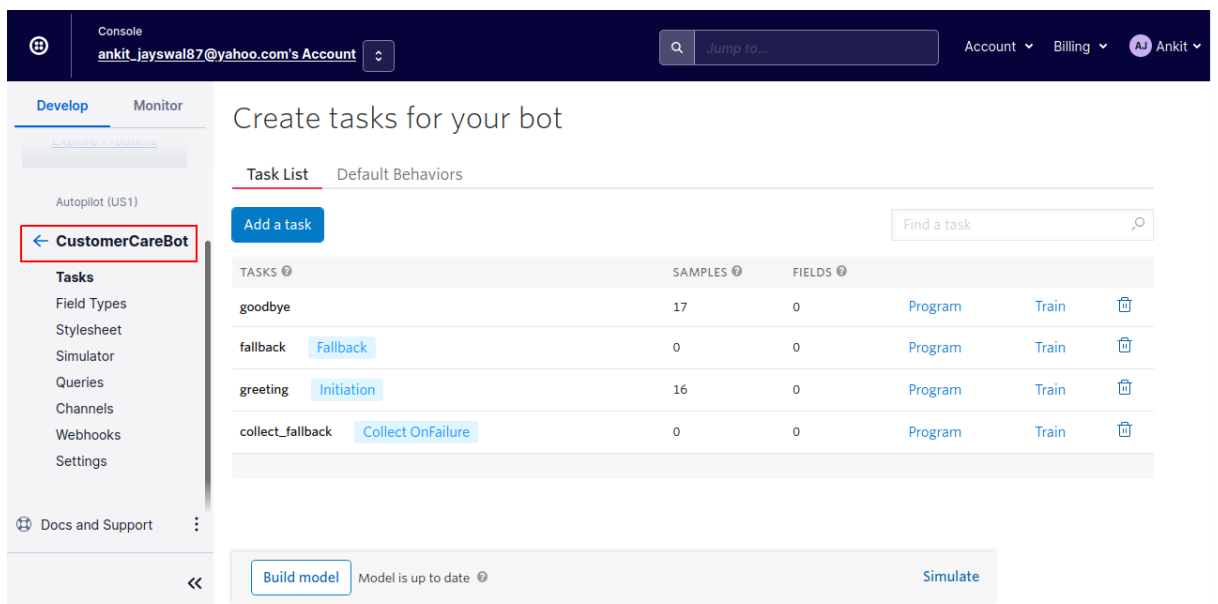


4) Name the Bot and Create

Here just give the name to your Bot "CustomerCareBot" and hit the create bot button.



- 5) You will now see the below screen, once you successfully created your first Bot. Here you can see the Bot with all its components which we mentioned and explained above in this book.



- 6) Create Tasks
Now let's create two simple tasks called **“services”** and **“timings”**
These tasks can answer customer queries related to services and timings of the company. So first we need to create these two tasks.

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

Autopilot (US1)

← CustomerCareBot

Tasks
Field Types
Stylesheet
Simulator
Queries
Channels
Webhooks
Settings

Docs and Support

Create tasks for your bot

Task List Default Behaviors

Add a task

Find a task

TASKS	SAMPLES	FIELDS			
goodbye	17	0	Program	Train	
fallback	0	0	Program	Train	
greeting	16	0	Program	Train	
collect_fallback	0	0	Program	Train	

Build model Model is up to date Simulate

Now name the task as **services** and click on Add button as per below.

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

Autopilot (US1)

← CustomerCareBot

Tasks
Field Types
Stylesheet
Simulator
Queries
Channels
Webhooks
Settings

Docs and Support

Create tasks for your bot

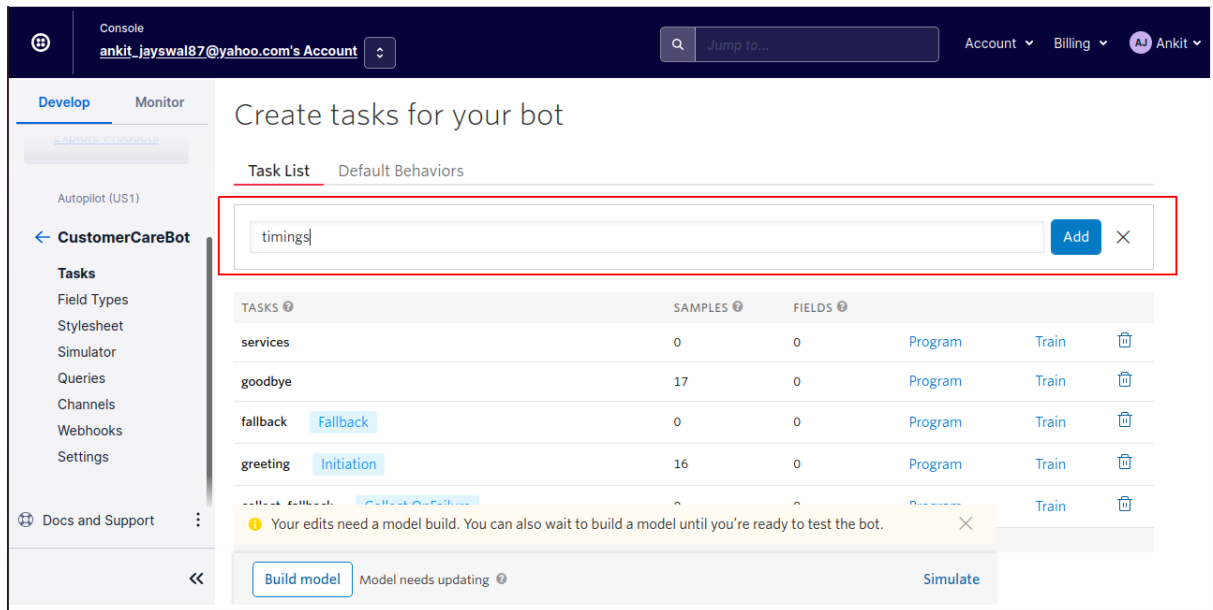
Task List Default Behaviors

services Add

TASKS	SAMPLES	FIELDS			
goodbye	17	0	Program	Train	
fallback	0	0	Program	Train	
greeting	16	0	Program	Train	
collect_fallback	0	0	Program	Train	

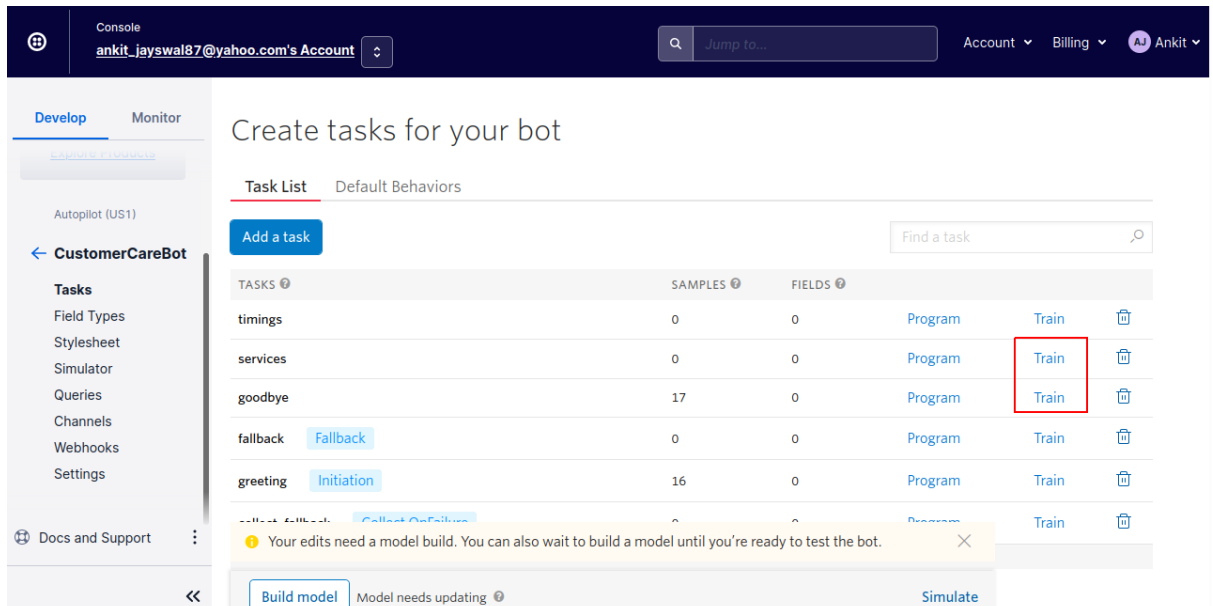
Build model Model is up to date Simulate

Likewise create **timings** tasks also as per below.

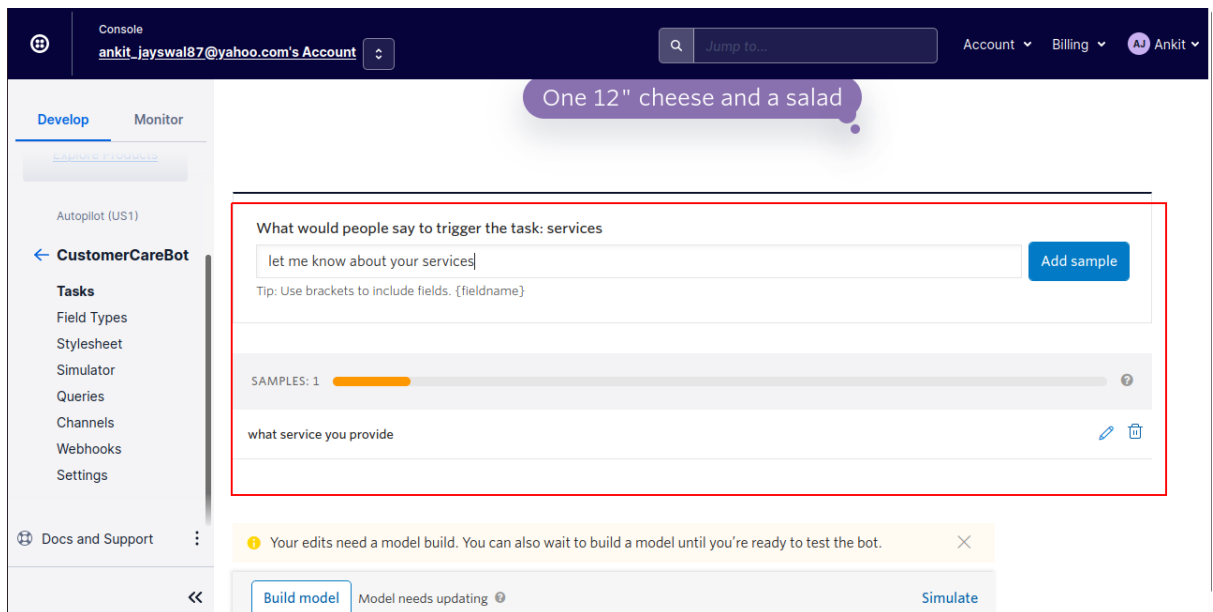


7) Train Tasks

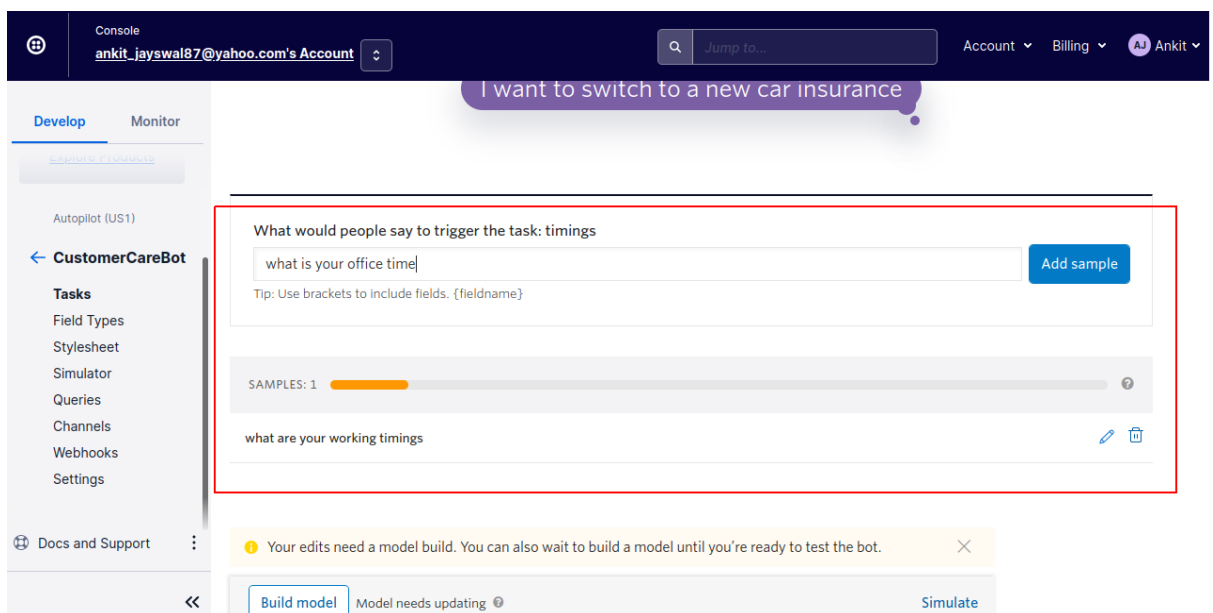
So, we have created two tasks called “**services**” and “**timings**”. Now I need to train these tasks. We will add sample data for both tasks to understand user input and to invoke these tasks. You can click on **Train** against each task to train and add sample data to it as shown below.



Adding samples for **services** task



Adding samples for **timings** task



8) Program Task

So, we have created tasks and trained the tasks. Now I need to program these tasks. Means when these tasks are invoked on user input what actual action needs to take. Here we will call simple **Say** Action, which will reply to the user with a given response.

Click on the **Program** as shown below.

Console
ankit_jayswal87@yahoo.com's Account

Account Billing Ankit

Develop Monitor

Autopilot (US1)

← CustomerCareBot

Tasks

Field Types
Stylesheet
Simulator
Queries
Channels
Webhooks
Settings

Docs and Support

Create tasks for your bot

Task List Default Behaviors

Add a task

Find a task

TASKS	SAMPLES	FIELDS			
timings	2	0	Program	Train	
services	2	0	Program	Train	
goodbye	17	0	Program	Train	
fallback	0	0	Program	Train	
greeting	16	0	Program	Train	

Your edits need a model build. You can also wait to build a model until you're ready to test the bot.

Build model Model needs updating Simulate

Program the **services** task as shown below and hit the save button.

Console
ankit_jayswal87@yahoo.com's Account

Account Billing Ankit

Develop Monitor

Autopilot (US1)

← CustomerCareBot

Tasks

Field Types
Stylesheet
Simulator
Queries
Channels
Webhooks
Settings

Docs and Support

UNIQUE NAME * services

FRIENDLY NAME ? Optional

Switch to train task

Program what the task does when triggered. ?

USE ☒ ActionBin ☐ Actions URL

```

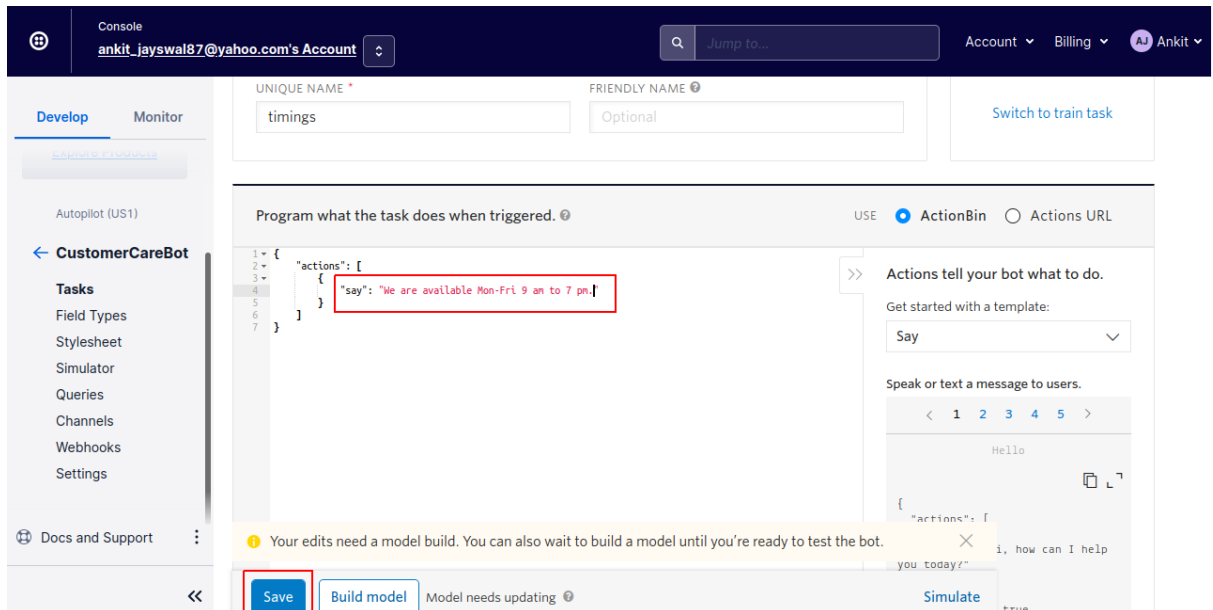
1 {
2   "actions": [
3     {
4       "say": "We are IT services and making solutions to help small businesses at affordable prices."
5     }
6   ]
7 }

```

Your edits need a model build. You can also wait to build a model until you're ready to test the bot.

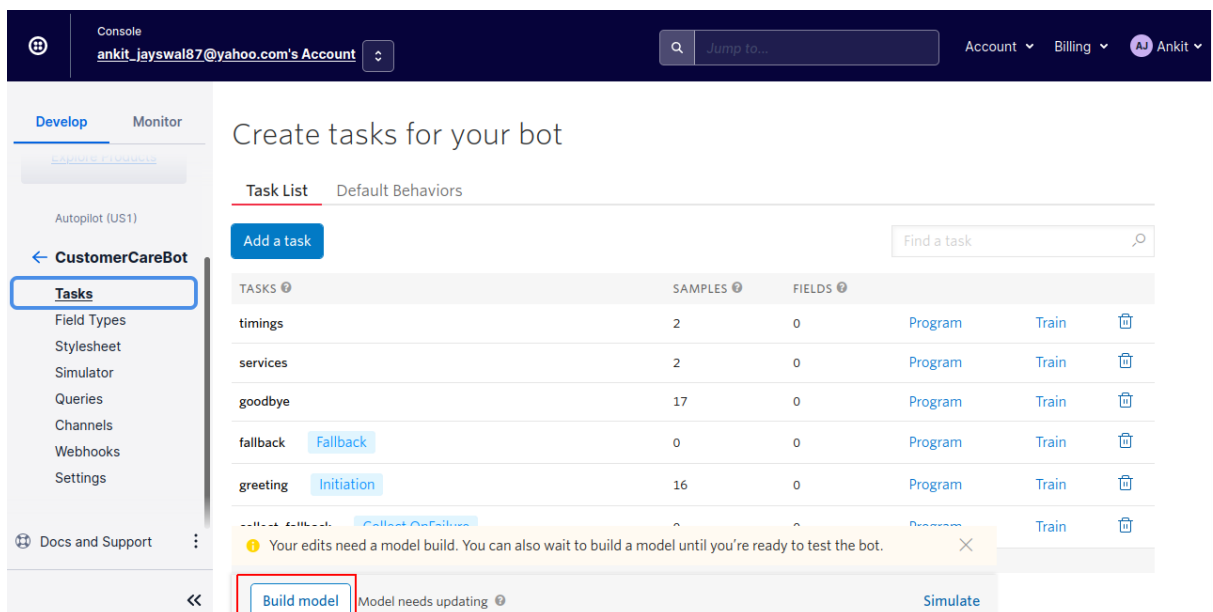
Save Build model Model needs updating Simulate

Program the **timings** task as shown below and hit the save button.



- 9) Great, now we have created tasks, trained tasks and provided proper Say Action to be executed when the task triggered. Now you might have noticed the **Build Model** button appeared in UI, this button appears whenever we add new tasks and train it with sample data. So, here Finally we need to Build a Model to take all the changes in effect.

Click on the Build model button as shown below in the image.



10) Test Bot in Simulator

Now we have successfully built our first model of **CustomerCareBot**. Let's test this Bot in a simulator. Click on Simulator as shown below in the image.

Building DataCollecting ChatBot

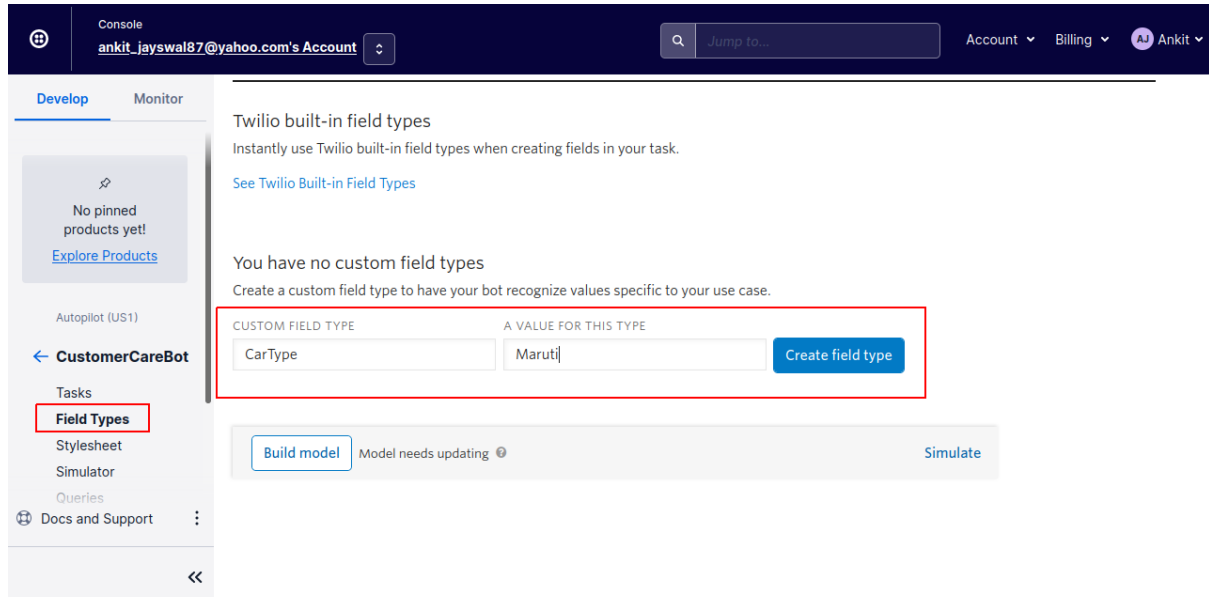
Start Building Data Collecting Bot

So above we learn how to make a simple conversational chatbot. It was answering static replies for the intent triggered by the user input.

Now to illustrate Data Collecting Bot, We will make one another Task of **Appointment Booking** into the same **CustomerCareBot**. This task will contain **collect** Action. And this task will ask basic questions to customers and collect all data. Once all mentioned questions are completed it will POST all data to specified URL. Here we will ask one Question regarding **CarType** the **custom Field** which will be validated against allowed values also.

So, let's start making the **Appointment Booking** task with collect Action. It will ask FirstName, LastName, PhoneNumber, CarType and Date of Booking. We will collect these answers on the specified URL in the same collect Action.

We can Create Custom Field **CarType** as below. Hit on the Create field type button as shown below.



Adding more values to the CarType field as below.

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

All Field Types > CarType

Add values for CarType

Tata

Add value

VALUES: 2

BMW

Maruti

SYNONYMS

Add a synonym

Add a synonym

Your edits need a model build. You can also wait to build a model until you're ready to test the bot.

Build model Model needs updating

Simulate

Below I am adding an Appointment Booking Task. Click on Add button as shown below.

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

Create tasks for your bot

Task List Default Behaviors

appointment_booking

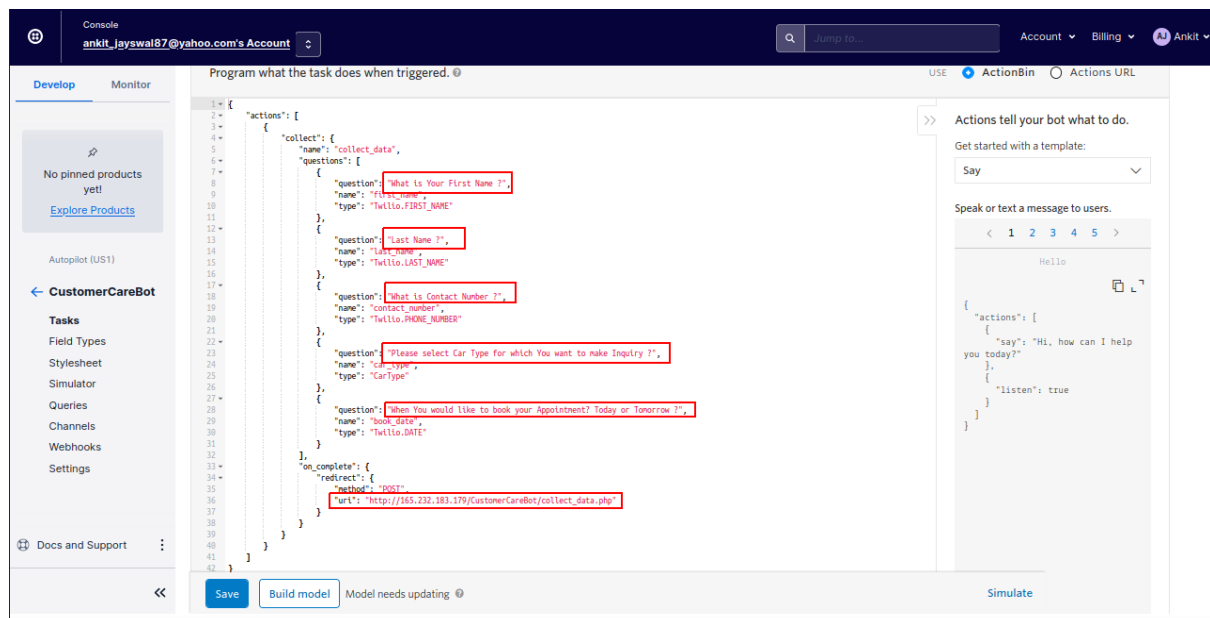
Add

TASKS	SAMPLES	FIELDS	Program	Train	
timings	2	0	Program	Train	
services	2	0	Program	Train	
goodbye	17	0	Program	Train	
fallback	0	0	Program	Train	
greeting	16	0	Program	Train	
collect_fallback	0	0	Program	Train	

Build model Model is up to date

Simulate

Now Program this task and provide collect Action as shown below in the image. Here you can see we asked various questions and at the end have specified one URL on which all data we will receive later. The URL can be any public url accessible over HTTP request.



The JSON code shown in above image, I am also sharing below to easily copy it. Below code will validate CarType value also. It will accept only Maruti, BMW or Tata car types only.

```
{
  "actions": [
    {
      "collect": {
        "name": "collect_data",
        "questions": [
          {
            "question": "What is Your First Name ?",
            "name": "first_name",
            "type": "Twilio.FIRST_NAME"
          },
          {
            "question": "Last Name ?",
            "name": "last_name",
            "type": "Twilio.LAST_NAME"
          },
          {
            "question": "What is Contact Number ?",
            "name": "contact_number",
            "type": "Twilio.PHONE_NUMBER"
          },
          {
            "question": "Please select Car Type for which You want to make Inquiry ?",
            "name": "car_type",
            "type": "CarType",
            "validate": {
              "on_failure": {
                "messages": [

```

```

        {
            "say": "Please select from
Maruti, BMW or Tata"
        },
        {
            "say": "We work for Maruti,
BMW or Tata. Please select from those."
        }
    ],
    "repeat_question": false
},
"max_attempts": {
    "redirect": "task://having-trouble",
    "num_attempts": 3
}
},
{
    "question": "When would you like to book your Appointment? Today
or Tomorrow ?",
    "name": "book_date",
    "type": "Twilio.DATE"
}
],
"on_complete": {
    "redirect": {
        "method": "POST",
        "uri": "http://165.232.183.179/CustomerCareBot/collect_data.php"
    }
}
}
}
]
}

```

The code at the web URL http://165.232.183.179/CustomerCareBot/collect_data.php is given below. This you can use for your reference for future development. The code mentioned below shows you how we can collect data from the `$_REQUEST['Memory']` parameter.

```

<?php

//collecting post data
$data = $_REQUEST['Memory'];
$data = json_decode($data,true);

$first_name = $data['twilio']['collected_data']['collect_data']['answers']['first_name']['answer'];
$last_name = $data['twilio']['collected_data']['collect_data']['answers']['last_name']['answer'];
$contact_number = $data['twilio']['collected_data']['collect_data']['answers']['contact_number']['answer'];
$car_type = $data['twilio']['collected_data']['collect_data']['answers']['car_type']['answer'];
$book_date = $data['twilio']['collected_data']['collect_data']['answers']['book_date']['answer'];

//You can store above received Data into choice of your Database via making DB connection here

$final_booking = "Okay, ".$first_name." ".$last_name.". We have Booked Your Appointment for CarType ".$car_type."
on ".$book_date.". Thank You for Booking with us.";

//producing JSON response to reply back
$redirectto = '{"actions": [{"say": "'.$final_booking.'"}]';
echo $redirectto;

?>

```

Now train the Appointment Booking task. Add sample data to invoke this task on user input. Click on the Add sample button as shown below. So here book appointment will be the keyword to invoke this task.

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

All Tasks > appointment_booking

0 Samples 0 Fields Switch to program task

WHAT ARE SAMPLES?

Samples are phrases that trigger your tasks.

I need a low interest rate CC.

What would people say to trigger the task: appointment_booking

book appointment Add sample

Tip: Use brackets to include fields. {fieldname}

Build model Model needs updating Simulate

Finally, don't forget to Build the model. Just click on Build model to take all changes in effect.

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

Create tasks for your bot

Task List Default Behaviors

Add a task Find a task

TASKS	SAMPLES	FIELDS			
timings	2	0	Program	Train	
services	2	0	Program	Train	
goodbye	17	0	Program	Train	
fallback	0	0	Program	Train	
greeting	16	0	Program	Train	
select fallback	0	0	Program	Train	

Your edits need a model build. You can also wait to build a model until you're ready to test the bot.

Build model Model needs updating Simulate

Okay now we are able to test this Bot for Appointment Booking task. We will go to the Simulator and test the Bot. Below are snapshots for the testing of Bot and Appointment Booking Done message coming from the webhook.

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

No pinned products yet!
[Explore Products](#)

Autopilot (US1)

← CustomerCareBot

Tasks
Field Types
Stylesheet
Simulator
Queries
Channels
Webhooks
Settings

Docs and Support

CustomerCareBot
Text simulator

Hi

Hello, what can I help you with today?

What is Your First Name ?

Last Name ?

What is Contact Number ?

Type a message

DEBUG VIEW

Start over

book appointment

Ankit

Jayswal

9979271234

"today"

Task Memory Dialogue Info

FIELD NAME	VALUE	TYPE
{last_name}	Jayswal	Twilio.LAST_NAME
{first_name}	Ankit	Twilio.FIRST_NAME
{contact_number}	9979271234	Twilio.PHONE_NUMBER
{car_type}	BMW	CarType
{book_date}	2022-01-03	Twilio.DATE

Build model Model is up to date

Console
ankit_jayswal87@yahoo.com's Account

Develop Monitor

No pinned products yet!
[Explore Products](#)

Autopilot (US1)

← CustomerCareBot

Tasks
Field Types
Stylesheet
Simulator
Queries
Channels
Webhooks
Settings

Docs and Support

CustomerCareBot
Text simulator

Please select Car Type for which You want to make Inquiry ?

Please select from Maruti, BMW or Tata

When You would like to book your Appointment? Today or Tomorrow ?

Type a message

DEBUG VIEW

Start over

Skoda

BMW

today

Okay, Ankit Jayswal. We have Booked Your Appointment for CarType BMW on 2022-01-03. Thank You for Booking with us.

"today"

Task Memory Dialogue Info

FIELD NAME	VALUE	TYPE
{last_name}	Jayswal	Twilio.LAST_NAME
{first_name}	Ankit	Twilio.FIRST_NAME
{contact_number}	9979271234	Twilio.PHONE_NUMBER
{car_type}	BMW	CarType
{book_date}	2022-01-03	Twilio.DATE

Build model Model is up to date

Facebook Channel Integration

Above we made CustomerCareBot and tested it within the simulator only. As I described above, this Bot we can integrate with various communications channels. Here we will discuss Facebook Channel Integration with the Bot. So the Bot can be used by Real Live Users over Facebook Messenger Application (User Interface).

Pre-requisites

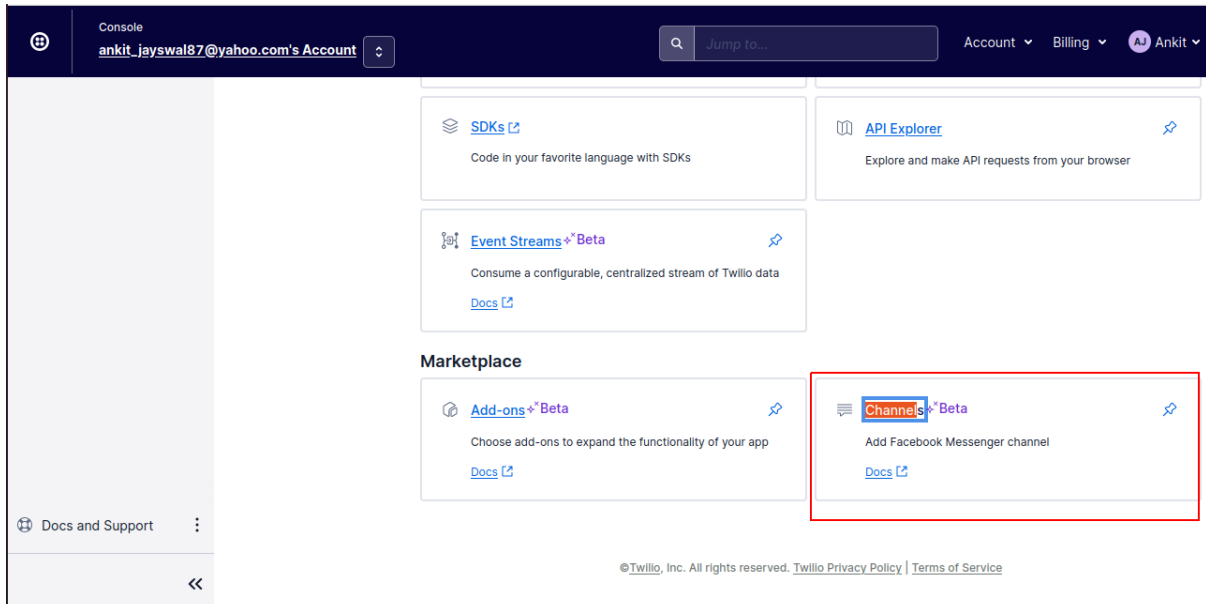
For this we need a working Facebook login account and then need to create one sample test Page on that Facebook account. This page will be the Identity of your Bot to the Real users. The live users will talk with this page and this page is integrated to our Bot. So users will get a reply back from our AI Bot for their queries. Users need Facebook Messenger App in their device, they can just search the page in it and start chatting with Bot.

I assume here you have a working Facebook login account and you have created a sample page on it. Also the Messenger App installed in the device to interact with ChatBot.

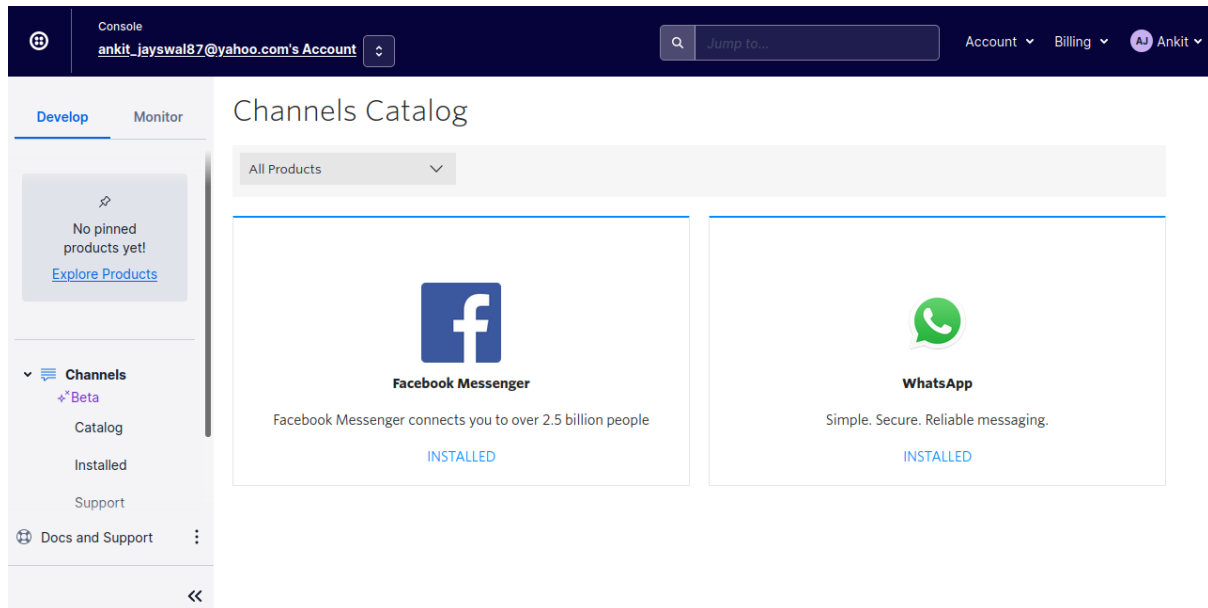
Process of Integration

Below I have mentioned required steps for Facebook channel integration with your Bot. So, carefully read the details below to make your integration successful and working.

Now, You need to go to Channels as shown below.

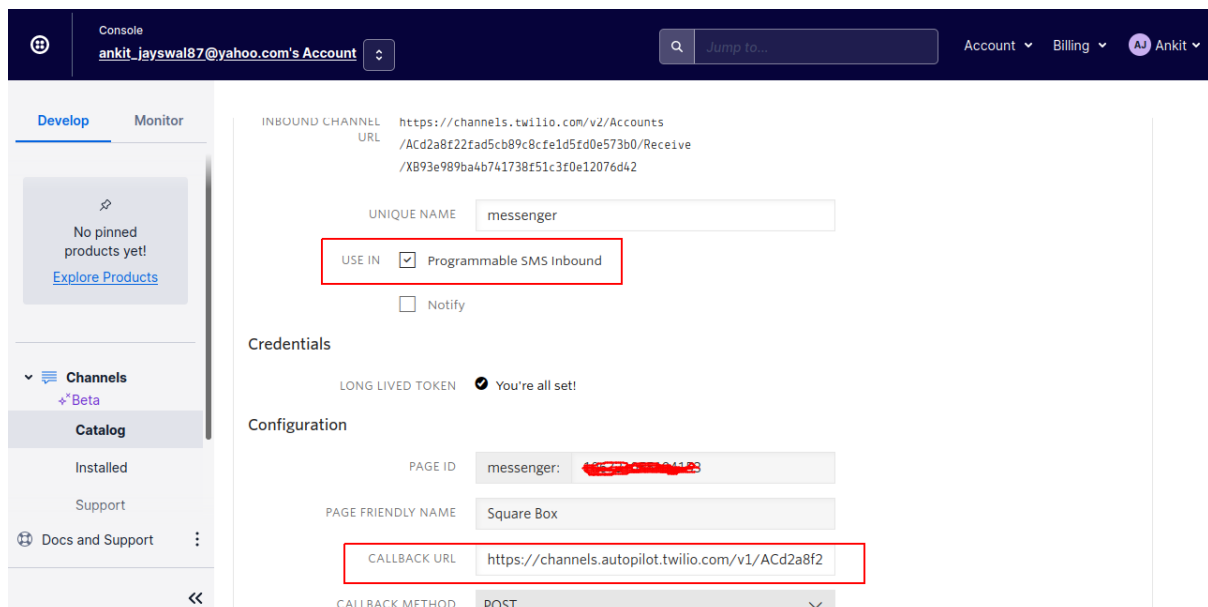


Once you click on Channels then you will get the page below. Here I already installed both Facebook and WhatsApp channels so it looks like this. Otherwise you need to click on the Install button.

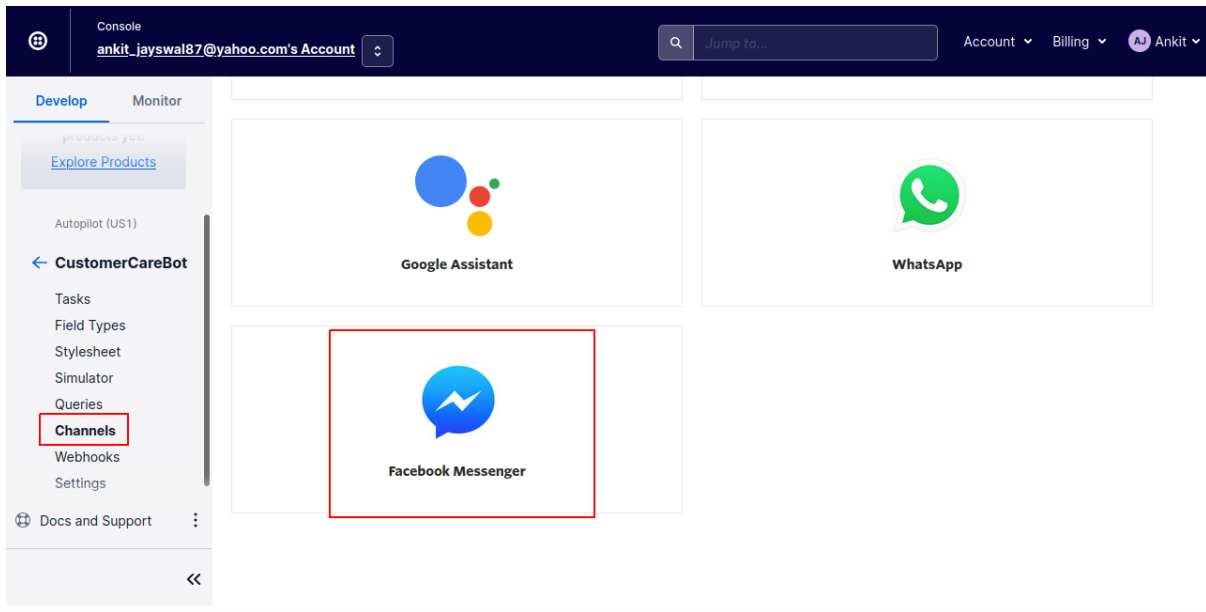


Okay fine so we have completed the pre-requisites. Just click on the above Facebook Messenger tile and you will see some configuration options. **Here you need to provide your Facebook Login details and you will be able to browse your created page just select it.** Then you will need to supply the **Callback URL** of your Bot that you will get from your Bot's channels tab.

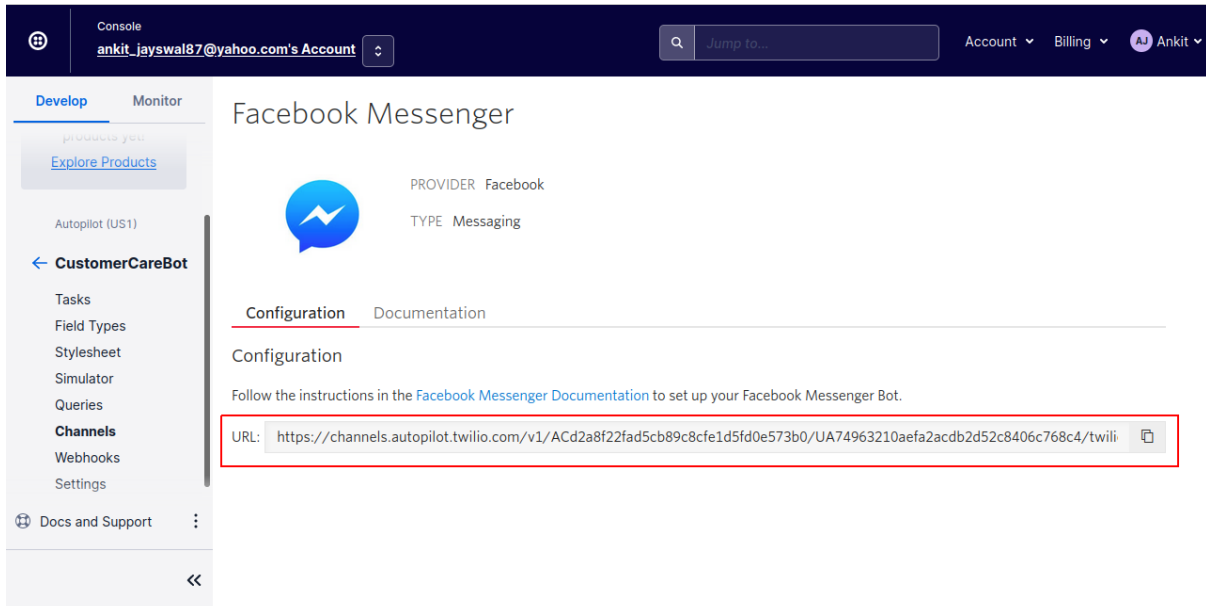
Just make a configuration like this as shown in the below image. **The Callback URL you will get from where I will explain to you.**



To get the **Callback URL**, just go to your bot. **Goto Autopilot -> Your bots -> Click on Your Bot -> Channels -> Facebook Messenger.** You will see the screen below. Click on Facebook Messenger.



Once you click on the Facebook Messenger tile you will see the screen below. Just copy this **FB URL (Callback URL)** of your Bot and paste it in the Above setting (Facebook Channel Setting) as per mentioned and save the setting. **So, now our installed FB Channel can communicate with the Bot.**



Nice, Now you will have to search your page in the Facebook Messenger app and try to chat with it. It should auto respond to you if all are set properly.

Congratulations! You have successfully integrated your Bot with Facebook Channel.

Business Ideas Around ChatBots

Nowadays chatbots are a very common interface of interacting. If you see at every website we find such a virtual assistant to answer our queries. If we provide chatbot services on FB and WhatsApp like worldwide channels people will use it and there will be a huge audience you can reach via these channels for your business.

Here I will discuss two most generic ideas which can be used in many businesses, services and in many sectors based on your imaginations. You can become a platform provider for the businesses for getting the Leads, Orders for them and become a promoter for the businesses of your choices. Below are some examples by which you will get a clear idea about it.

List of Ideas

So, let's have some good examples, on which I have worked.

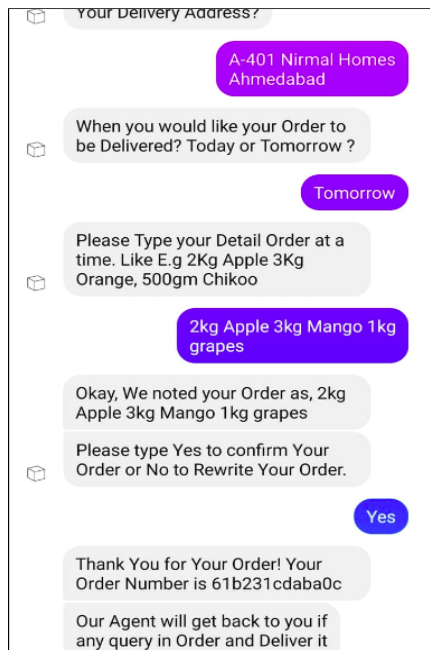
1. Create Revenue generating channels

You can create an Order taking interface for any kind of business and approach Business Owners for using it. You can charge a small recurring Platform Fee or commission on every Order. Here you can also work as a promoter for that Business. More the Orders, more Commission.

Example:-

- Grocery, Fruits or Vegetables Order taking Interface over FB channel
- Providing Order taking platform to Vendor and charge commission per Order
- You can charge small platform fee to use it

Below is the Fruit Order Booking example over Facebook Messenger.



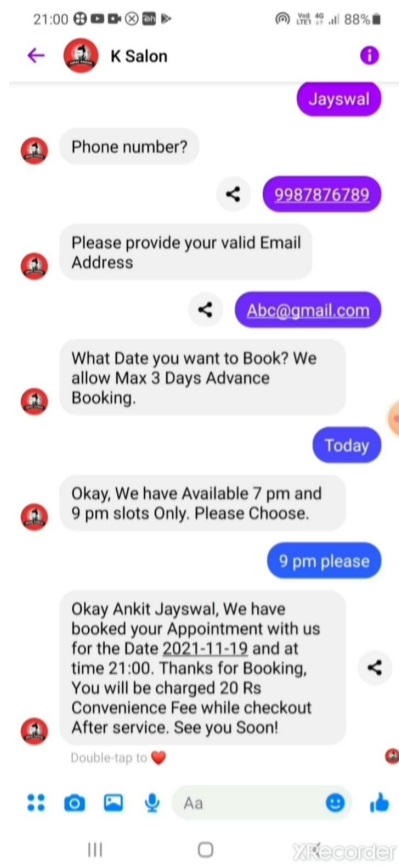
2. Time Saving Application

You can create Appointment Booking ChatBot Application for many services. Like Salon Appointment Booking Application. Here you can provide Daily 2 or 3 Premium bookings. These Booking slots can be used by customers to save their valuable time at Salon and they get served in a timely manner. For such premium Bookings, customers can pay some extra convenience Fee to Owner.

Likewise you can make this Appointment Booking application for Doctors also. Like allowing only 15 Patients for the Day. Bookings more than 15 will not be allowed and it will say, "Bookings are full for Today, Please book your appointment for Tomorrow." This way the Organisation knows in advance, how many patients need to be served by them today. And the Receptionist will not have to book appointments manually over the phone every time.

Likewise the same can be applied to book a table in the Restaurants. People can pay for their time saving and they get served the Food in a timely manner.

Below is the Salon Appointment Booking Example.



Demo Videos of Real Projects

Below are some of my Real projects video links available to watch. Once you integrate database connection and API connections in development you will have much more Dynamic and custom flavour of ChatBot responses.

Video Links

Coffee Bot on WhatsApp:

<https://drive.google.com/file/d/1bJM5ym6rKfbqgHiwVwF-ljlrw20cWr3R/view?usp=sharing>

Salon Appointment Booking on Facebook Messenger:

<https://drive.google.com/file/d/1Sf6IkRbqJ5DnigbzSjOR7EBjQVYQ8QWX/view?usp=sharing>

Audio Links

Pizza Order VoiceBot Recording (future order):

<https://drive.google.com/file/d/1PatHfD756VSILDlgzp40MG1ea9pLIK6u/view?usp=sharing>

Pizza Order VoiceBot Recording (order now):

https://drive.google.com/file/d/1aRM10V648qIZrehK9fnAKpc_XFAMSi0v/view?usp=sharing

**Thank
You**