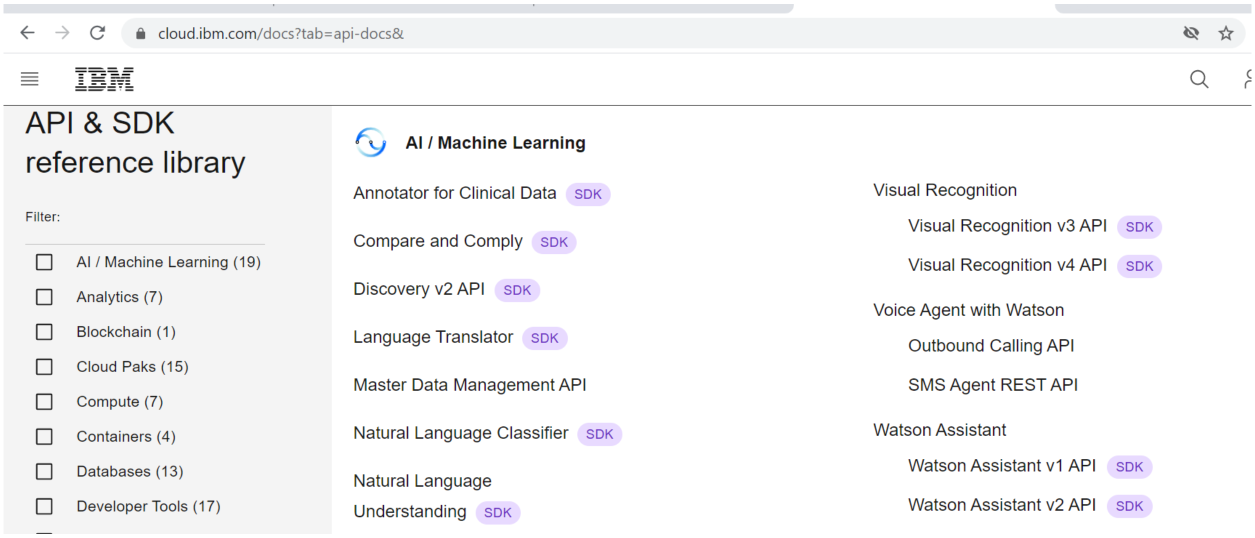
Step 1: open cloud.ibm.com/docs

Step 2: select Watson Assistant v2 API



API & SDK reference library

Filter:

- □ AI / Machine Learning (19)
- □ Analytics (7)
- □ Blockchain (1)
- □ Cloud Paks (15)
- □ Compute (7)
- □ Containers (4)
- □ Databases (13)
- □ Developer Tools (17)

AI / Machine Learning

Annotator for Clinical Data SDK

Compare and Comply SDK

Discovery v2 API SDK

Language Translator SDK

Master Data Management API

Natural Language Classifier SDK

Natural Language Understanding SDK

Visual Recognition

- Visual Recognition v3 API SDK
- Visual Recognition v4 API SDK

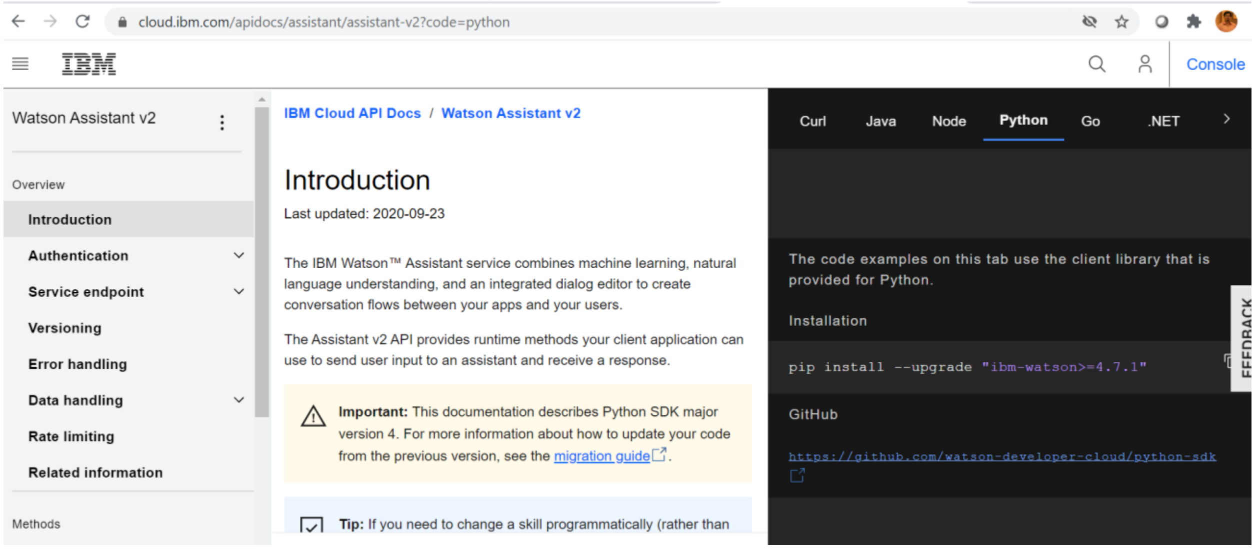
Voice Agent with Watson

- Outbound Calling API
- SMS Agent REST API

Watson Assistant

- Watson Assistant v1 API SDK
- Watson Assistant v2 API SDK

Step 3: Install below shown library by coping



IBM Cloud API Docs / Watson Assistant v2

Introduction

Last updated: 2020-09-23

The IBM Watson™ Assistant service combines machine learning, natural language understanding, and an integrated dialog editor to create conversation flows between your apps and your users.

The Assistant v2 API provides runtime methods your client application can use to send user input to an assistant and receive a response.

Important: This documentation describes Python SDK major version 4. For more information about how to update your code from the previous version, see the migration guide.

Tip: If you need to change a skill programmatically (rather than

The code examples on this tab use the client library that is provided for Python.

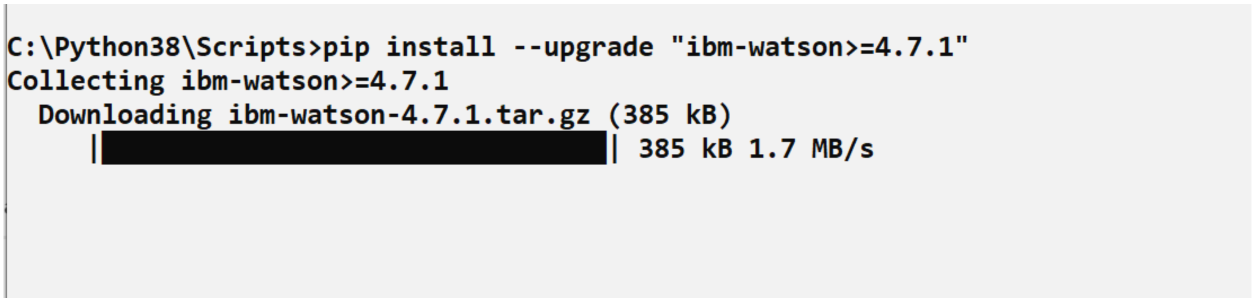
Installation

```
pip install --upgrade "ibm-watson>=4.7.1"
```

GitHub

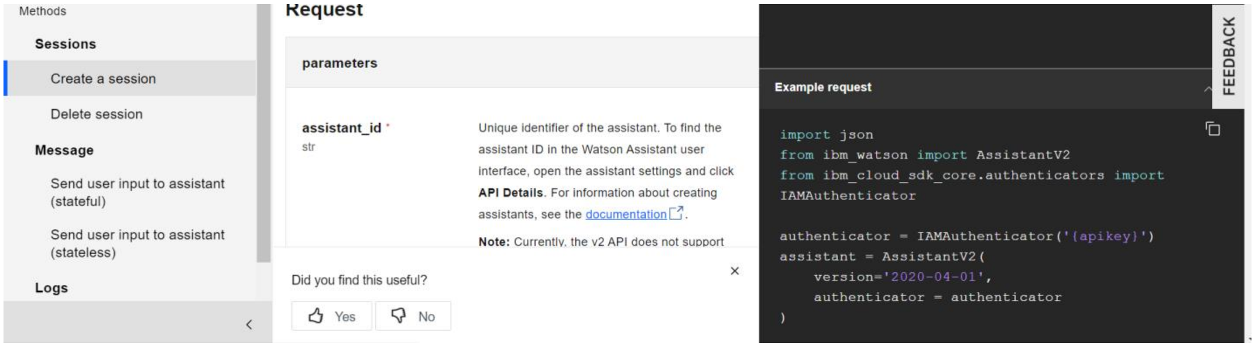
https://github.com/watson-developer-cloud/python-sdk

Step 4: goto cmd, python path and paste like below



```
C:\Python38\Scripts>pip install --upgrade "ibm-watson>=4.7.1"
Collecting ibm-watson>=4.7.1
  Downloading ibm-watson-4.7.1.tar.gz (385 kB)
     |████████████████████████████████| 385 kB 1.7 MB/s
```

Step 5: Create a session



Methods
Sessions
Create a session
Delete session
Message
Send user input to assistant (stateful)
Send user input to assistant (stateless)
Logs
Request
parameters
Did you find this useful?
Yes
No
Example request
FEEDBACK

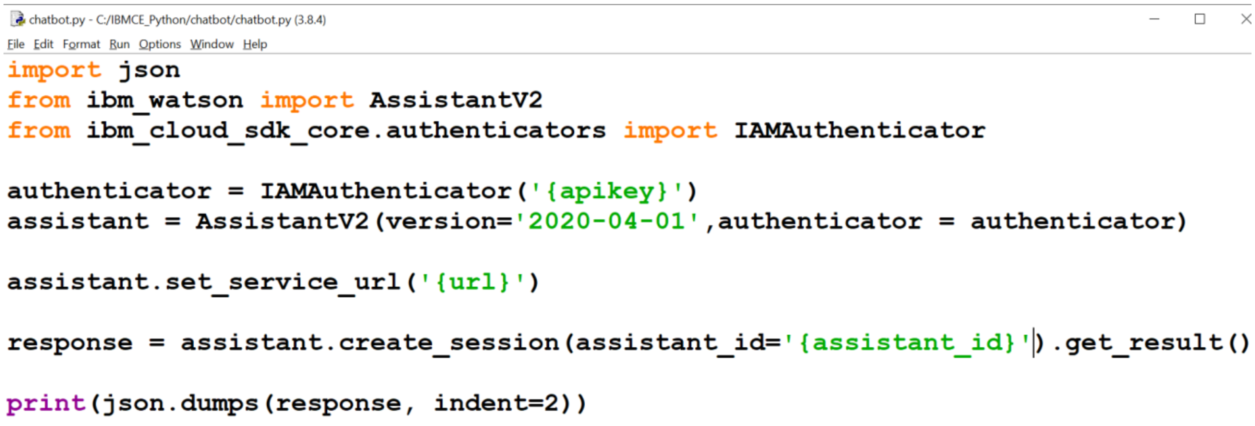
assistant_id *
str

Unique identifier of the assistant. To find the assistant ID in the Watson Assistant user interface, open the assistant settings and click **API Details**. For information about creating assistants, see the documentation.

Note: Currently, the v2 API does not support

```
import json
from ibm_watson import AssistantV2
from ibm_cloud_sdk_core.authenticators import
IAMAuthenticator

authenticator = IAMAuthenticator('{apikey}')
assistant = AssistantV2(
    version='2020-04-01',
    authenticator = authenticator
)
```

Step 6: copy the RHS code into a python file

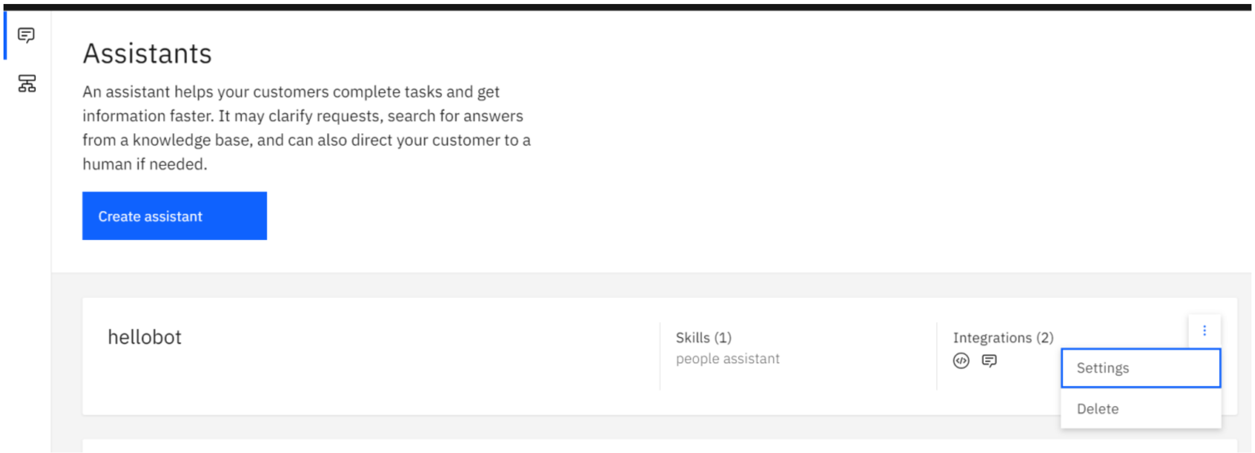


chatbot.py - C:/IBMCE_Python/chatbot/chatbot.py (3.8.4)
File Edit Format Run Options Window Help

```
import json
from ibm_watson import AssistantV2
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

authenticator = IAMAuthenticator('{apikey}')
assistant = AssistantV2(version='2020-04-01',authenticator = authenticator)

assistant.set_service_url('{url}')

response = assistant.create_session(assistant_id='{assistant_id}').get_result()

print(json.dumps(response, indent=2))
```

Step 7: Replace apikey, url and assistant_id with actual values

Step 8: Goto Assistant settings -> API Details -> Assistant ID



Assistants

An assistant helps your customers complete tasks and get information faster. It may clarify requests, search for answers from a knowledge base, and can also direct your customer to a human if needed.

Create assistant

hellobot

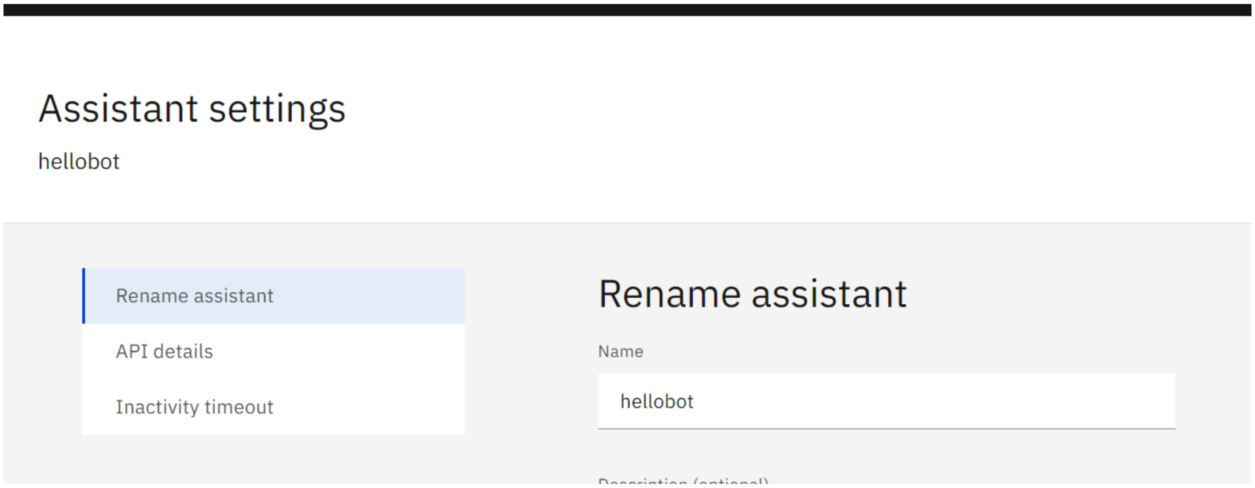
Skills (1)
people assistant

Integrations (2)

Settings

Delete

Step 9: Click on API details



Assistant settings

hellobot

Rename assistant

API details

Inactivity timeout

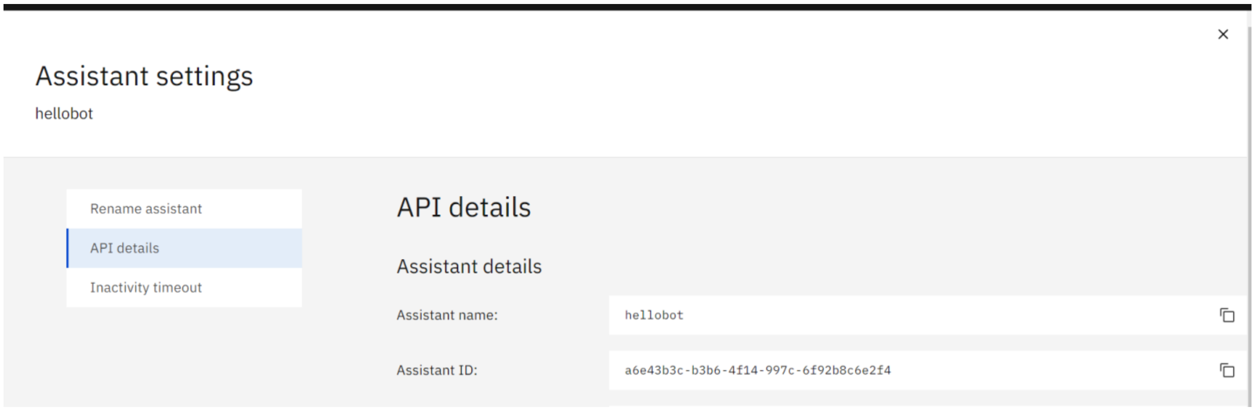
Rename assistant

Name

hellobot

Description (optional)

Step 10: Copy the Assistant ID and paste in Python code



Assistant settings

hellobot

Rename assistant

API details

Inactivity timeout

API details

Assistant details

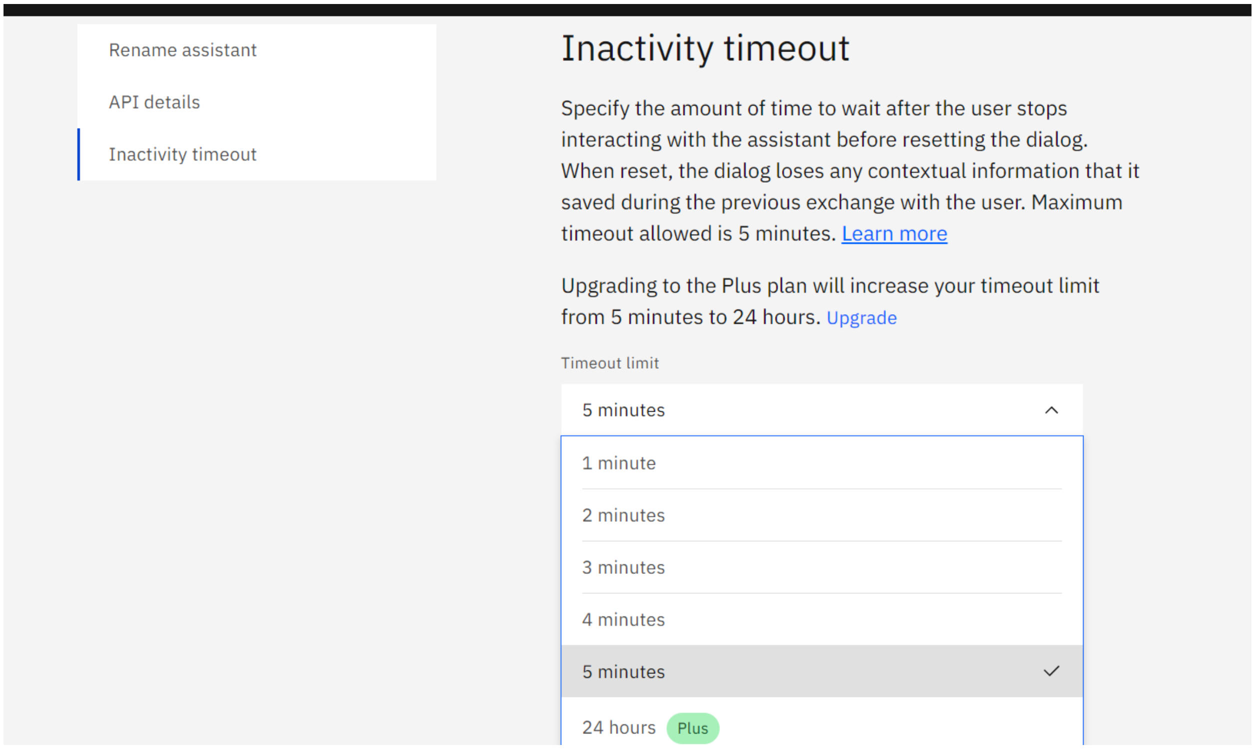
Assistant name: hellobot

Assistant ID: a6e43b3c-b3b6-4f14-997c-6f92b8c6e2f4

Session gets delete for every certain period

Step 11: Click on Inactivity timeout and understand the Timeout limit

Note: Because our accounts are free accounts we cannot increase the time limit



Inactivity timeout

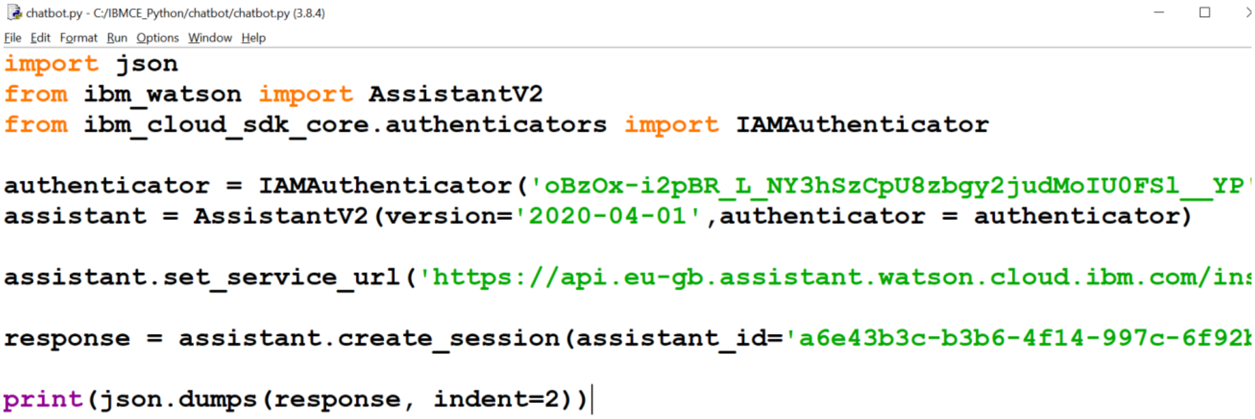
Specify the amount of time to wait after the user stops interacting with the assistant before resetting the dialog. When reset, the dialog loses any contextual information that it saved during the previous exchange with the user. Maximum timeout allowed is 5 minutes. Learn more

Upgrading to the Plus plan will increase your timeout limit from 5 minutes to 24 hours. Upgrade

Timeout limit

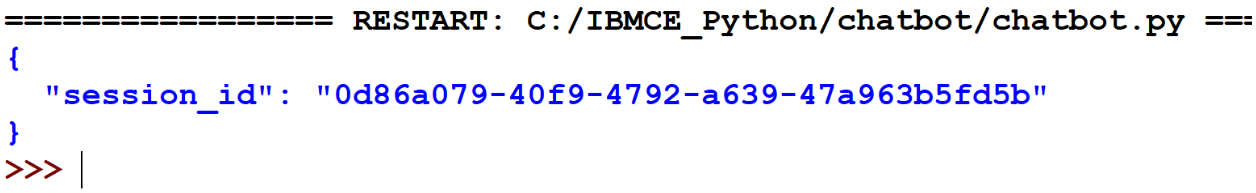
5 minutes

Step 12: Run the below code by click f5


chatbot.py - C:/IBMCE_Python/chatbot/chatbot.py (3.8.4)
File Edit Format Run Options Window Help

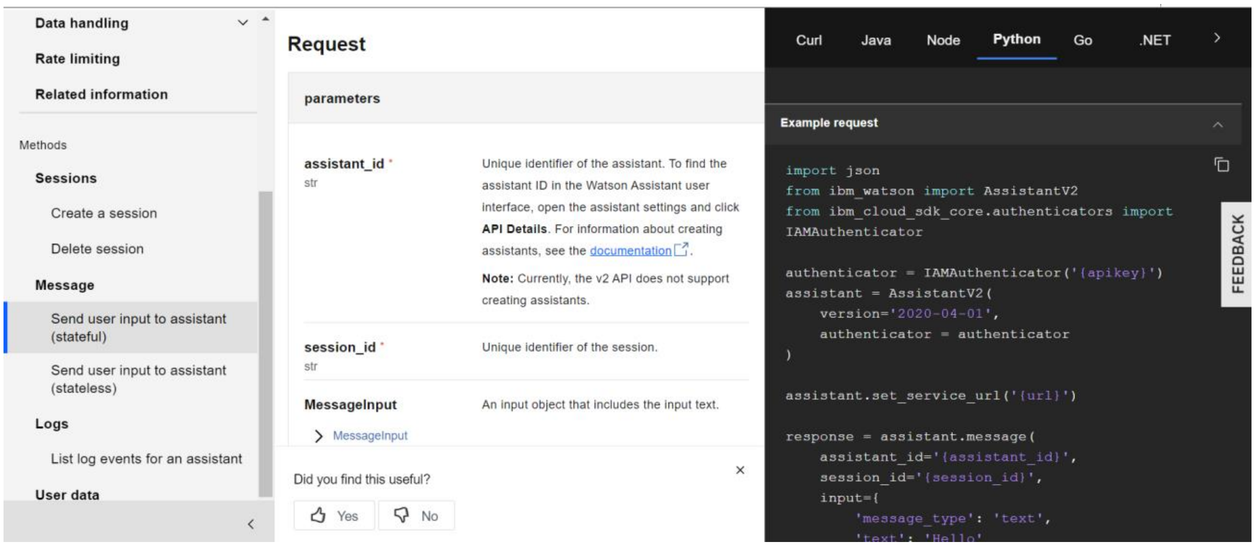
```
import json
from ibm_watson import AssistantV2
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

authenticator = IAMAuthenticator('oBzOx-i2pBR_L_NY3hSzCpU8zbgy2judMoIU0FSl__YP'
assistant = AssistantV2(version='2020-04-01',authenticator = authenticator)

assistant.set_service_url('https://api.eu-gb.assistant.watson.cloud.ibm.com/ins

response = assistant.create_session(assistant_id='a6e43b3c-b3b6-4f14-997c-6f92

print(json.dumps(response, indent=2))
```

Step 13: Ensure output should be like below with session id, copy and paste in notepad



```
================= RESTART: C:/IBMCE_Python/chatbot/chatbot.py ===
{
  "session_id": "0d86a079-40f9-4792-a639-47a963b5fd5b"
}
>>> 
```

Step 14: select Send user input to assistant (statefull), copy the code



Request

parameters

assistant_id * str — Unique identifier of the assistant. To find the assistant ID in the Watson Assistant user interface, open the assistant settings and click **API Details**. For information about creating assistants, see the documentation.

Note: Currently, the v2 API does not support creating assistants.

session_id * str — Unique identifier of the session.

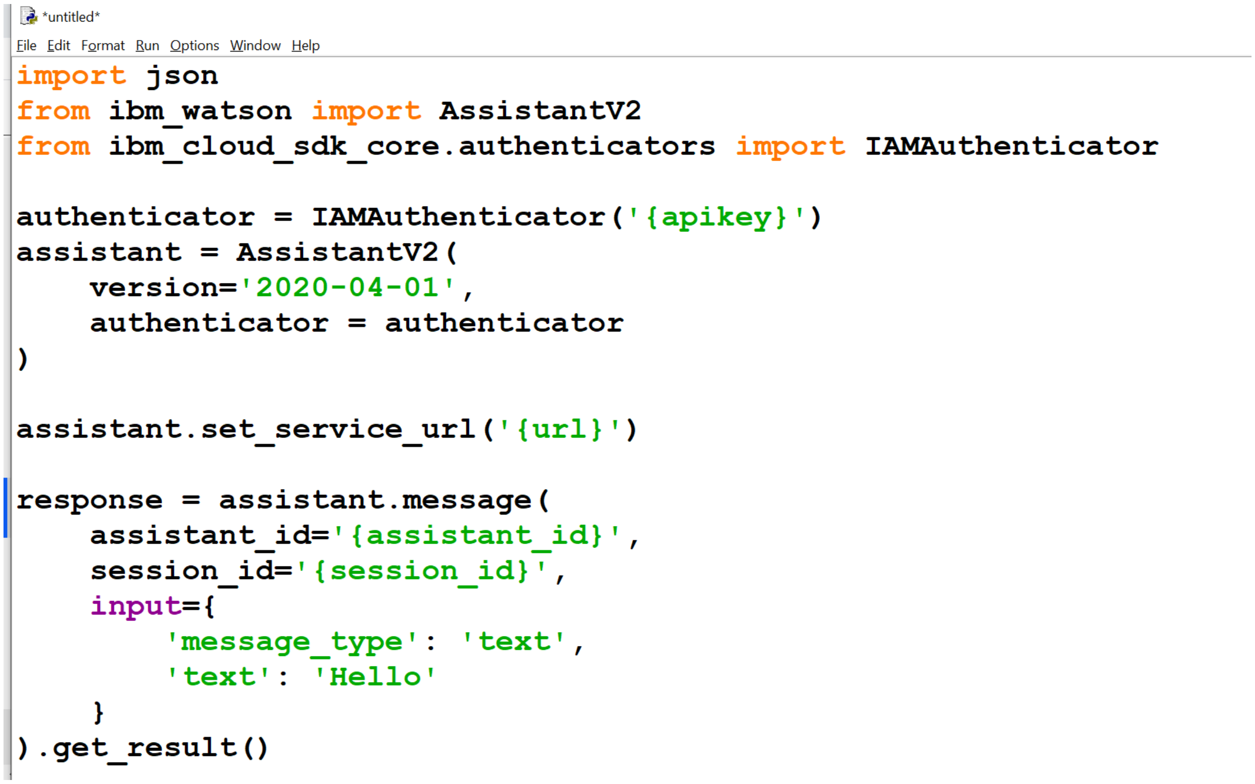
MessageInput — An input object that includes the input text.

MessageInput

Example request

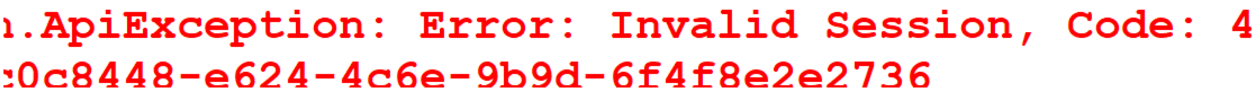
```
import json
from ibm_watson import AssistantV2
from ibm_cloud_sdk_core.authenticators import
IAMAuthenticator

authenticator = IAMAuthenticator('{apikey}')
assistant = AssistantV2(
    version='2020-04-01',
    authenticator = authenticator
)

assistant.set_service_url('{url}')

response = assistant.message(
    assistant_id='{assistant_id}',
    session_id='{session_id}',
    input={
        'message_type': 'text',
        'text': 'Hello'
```

Step 15: Paste the code in new python file



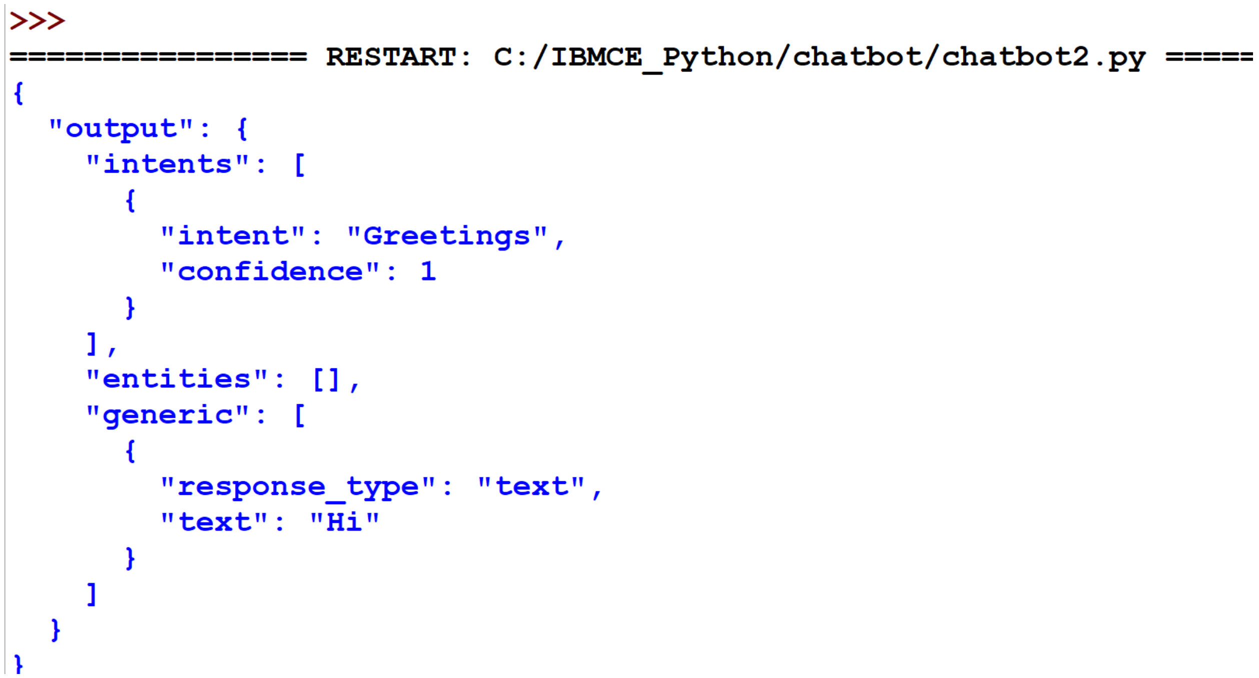
```
import json
from ibm_watson import AssistantV2
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

authenticator = IAMAuthenticator('{apikey}')
assistant = AssistantV2(
    version='2020-04-01',
    authenticator = authenticator
)

assistant.set_service_url('{url}')

response = assistant.message(
    assistant_id='{assistant_id}',
    session_id='{session_id}',
    input={
        'message_type': 'text',
        'text': 'Hello'
    }
).get_result()
```

Step 16: Replace 4 details (session id extra got included) and execute the code you will see below error



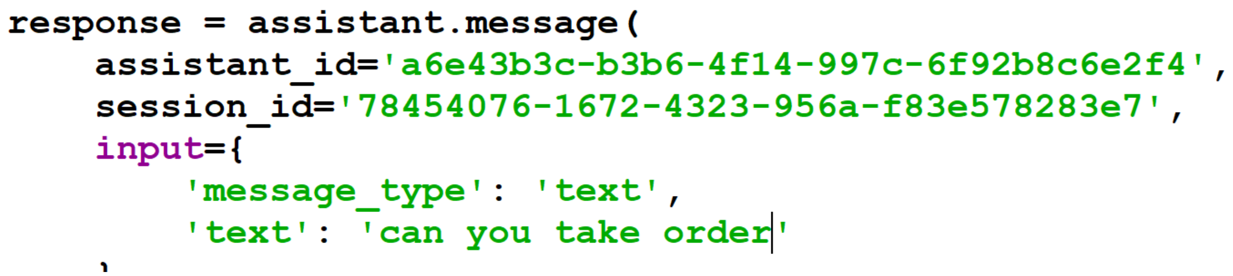
```
.ApiException: Error: Invalid Session, Code: 4
0c8448-e624-4c6e-9b9d-6f4f8e2e2736
```

Step 17: Re execute previoud python file so that you will get output with latest session id, copy that new session id in above code and execute. You will get the output like below



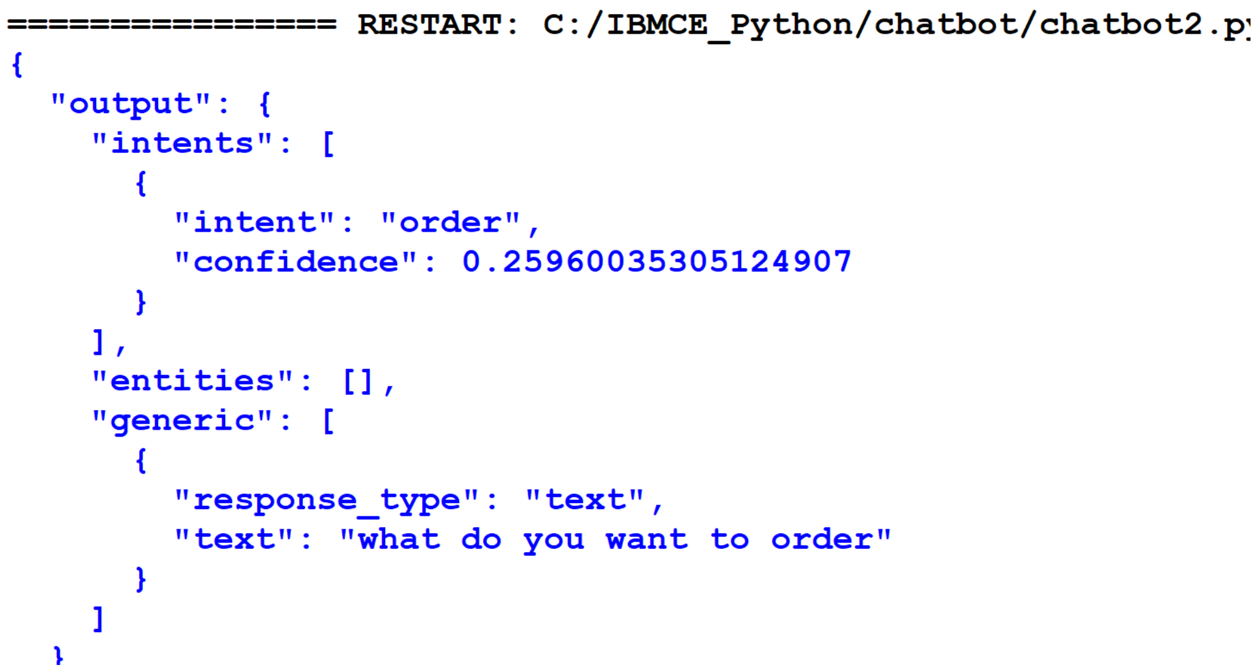
```
>>> 
================ RESTART: C:/IBMCE_Python/chatbot/chatbot2.py =====
{
  "output": {
    "intents": [
      {
        "intent": "Greetings",
        "confidence": 1
      }
    ],
    "entities": [],
    "generic": [
      {
        "response_type": "text",
        "text": "Hi"
      }
    ]
  }
}
```

Step 18: change input to “can you take order”



```
response = assistant.message(
    assistant_id='a6e43b3c-b3b6-4f14-997c-6f92b8c6e2f4',
    session_id='78454076-1672-4323-956a-f83e578283e7',
    input={
        'message_type': 'text',
        'text': 'can you take order'
```

Step 19: Now execute code you can see result like below



```
================ RESTART: C:/IBMCE_Python/chatbot/chatbot2.py
{
  "output": {
    "intents": [
      {
        "intent": "order",
        "confidence": 0.25960035305124907
      }
    ],
    "entities": [],
    "generic": [
      {
        "response_type": "text",
        "text": "what do you want to order"
      }
    ]
  }
```