Doctor Bot Sample

Overview

This article will demonstrate how to create a bot to detect your sickness.

Requirements

- Python 3.6
- Bot Builder SDK v4 (Python Library)
- Bot Framework Emulator

Scenario

If you're not feeling good, and think you have a disease, then try to talk with the bot all the symptoms you have, and the bot will access it's database and tell about your disease based on the symptoms.

Getting Started

First of all you need to install your **Python 3.6** in your machine. You download it directly from: https://www.python.org/downloads/release/python-360/

After that, you need to set up for **Bot Builder v4** environment by installing **Bot Builder SDK V4** python library. You can install it with using pip3 command: ***pip install botbuilder-core==4.4.0b1*** or you can download the package manually from https://pypi.org/project/botbuilder-core/4.4.0b1/ and install it in your python3 library.

After that you can use a sample bot template from:
https://github.com/Microsoft/botbuilder-python/tree/master/samples
We should have all we need to get started now. If you need a working example, feel free to download my sample bot project at https://github.com/juanchrisniady/DoctorBot

For this bot, the first step is to set up a server and port of the bot.

```
try:
    print('Starting server on: ' + 'http://localhost:' + str(PORT))
    SERVER = http.server.HTTPServer(('localhost', PORT), BotRequestHandler)
    SERVER.serve_forever()
except KeyboardInterrupt:
    SERVER.socket.close()
```

And then, we need to have some methods/functions for the bot to handle any activity in the server.

```
  def do_POST(self):
      body = self.rfile.read(int(self.headers['Content-Length']))
      data = json.loads(str(body, 'utf-8'))
      activity = Activity.deserialize(data)

      if not self.__handle_authentication(activity):
          return

      # Handle when someone join the server
      if activity.type == ActivityTypes.conversation_update.value:
          self.__handle_conversation_update_activity(activity)
      # handle when someone write a message on server
      elif activity.type == ActivityTypes.message.value:
          self.__handle_message_activity(activity)
      # handle unknown error
      else:
          self.__unhandled_activity()
```

Next, we need to setup the function for the bot to introduce itself and and reply to a message:

```
def __create_reply_activity(request_activity, text):
    return Activity(
        type=ActivityTypes.message,
        channel_id=request_activity.channel_id,
        conversation=request_activity.conversation,
        recipient=request_activity.from_property,
        from_property=request_activity.recipient,
        text=text,
        service_url=request_activity.service_url)

def __handle_conversation_update_activity(self, activity):
    self.send_response(202)
    self.end_headers()
    if activity.members_added[0].id != activity.recipient.id:
        credentials = MicrosoftAppCredentials(APP_ID, APP_PASSWORD)
        reply = BotRequestHandler.__create_reply_activity(activity, 'Hello, welcome to Doctor Bot.\n'
                                                          + 'If you are not feeling well, tell me whats wrong, '
                                                          + 'and I will try to help you out')
        connector = ConnectorClient(credentials, base_url=reply.service_url)
        connector.conversations.send_to_conversation(reply.conversation.id, reply)

def __handle_message_activity(self, activity):
    self.send_response(200)
    self.end_headers()
    credentials = MicrosoftAppCredentials(APP_ID, APP_PASSWORD)
    connector = ConnectorClient(credentials, base_url=activity.service_url)
    reply = BotRequestHandler.__create_reply_activity(activity, BotRequestHandler.predict_illness(activity.text))
    connector.conversations.send_to_conversation(reply.conversation.id, reply)

def __handle_authentication(self, activity):
    credential_provider = SimpleCredentialProvider(APP_ID, APP_PASSWORD)
    loop = asyncio.new_event_loop()
    try:
        loop.run_until_complete(JwtTokenValidation.authenticate_request(
            activity, self.headers.get("Authorization"), credential_provider))
        return True
    except Exception as ex:
        self.send_response(401, ex)
        self.end_headers()
        return False
    finally:
        loop.close()
```

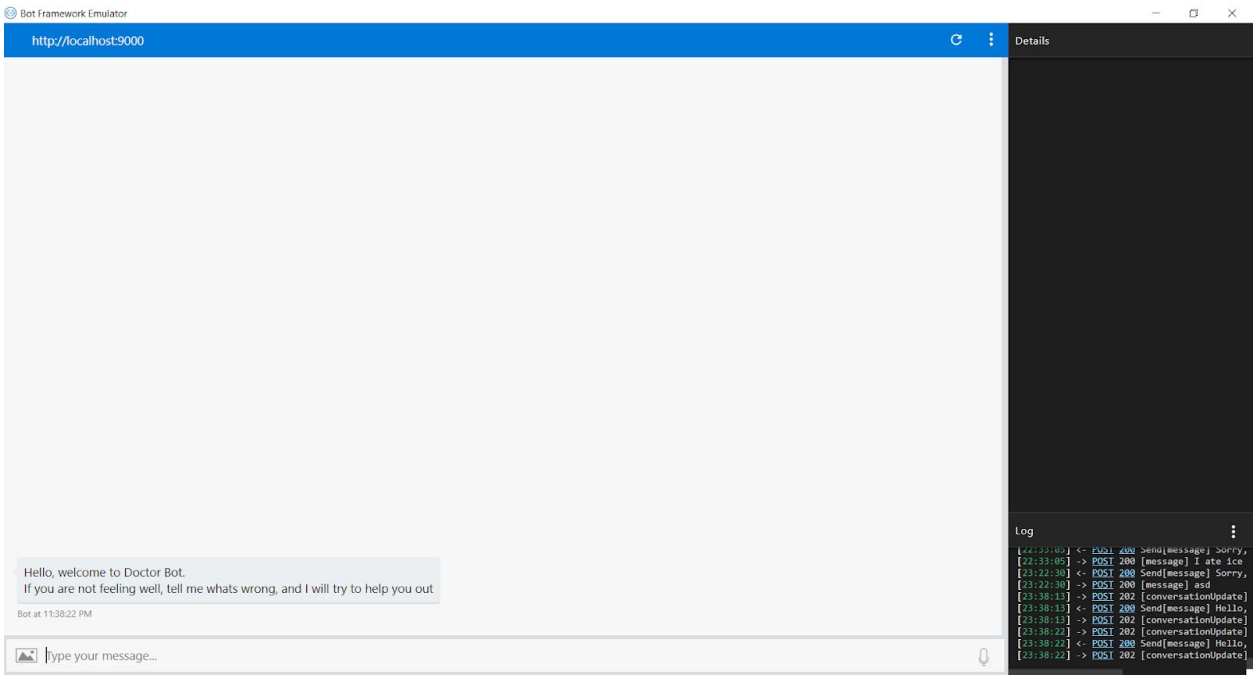
At this point, the bot should be able to recognize when someone is joining the server or send a message on the server. Now, we need to prepare some data of disease and their symptoms.

```
# these are diseases and their symptoms
f_symptoms = ['fever', 'cough', 'sore throat', 'runny', 'nose', 'muscle', 'headaches', 'fatigue']
d_symptoms = ['loose', 'watery', 'stools', 'cramps', 'fever', 'blood', 'bloating', 'nausea', 'bowel']
```

And the last step is to make a function to predict a disease based on the word count of a disease symptom in the message.

```python
def predict_illness(self, str):
    f_count = 0 #flu symthoms count
    d_count = 0 #diarhea symthoms count
    for fs in self.f_symptoms:
        if fs in str:
            f_count += 1
    for ds in self.d_symptoms:
        if ds in str:
            d_count += 1
    if(f_count > d_count):
        return 'You might have a flu.\n' + 'Check this link for a Flu treatment: https://www.mayoclinic.org/diseases-conditions/flu/diagnosis-treatment/drc-20351725'
    elif(d_count > f_count):
        return 'You might have a diarrhea.\n' + 'Check this link for a Diarrhea treatment: https://www.emedicinehealth.com/diarrhea/article_em.htm'
    else:
        return 'Sorry, we dont really know your illness based on your synthoms'
```

Now to use the bot, you can start your bot (use command ***python mainbot.py*** if you're using my example), and then open Bot Framework Emulator and connect it to http://localhost:9000 , and you should see this screen when your bot is successfully running.



Conclusion

This tutorial shows how to create a simple doctor bot using Bot Builder V4 (python) and Bot Framework Emulator.