COMP7940 Cloud Computing

2022/23 S2 Lab 3 Starting a TG chatbot

Instructor	Dr. Qichen Wang	qcwang@hkbu.edu.hk
Teaching Assistant	Mr. Zhengheng Tang	zhtang@comp.hkbu.edu.hk

Objective:

Throughout this lab you will be able to: 1. Experience in running a chatbot; 2. Customize your chatbot (greeting message, issue special command);

---

1. Installation

Install the following software on your phone:

- Telegram

2. Start a chatbot

Telegram provides an official Bots accounts that do not require additional phone number to set up. You can add @BotFather and just talk to the @BotFather follow some simple step to create your own chatbot.

you can find more about the Telegram bot API at: https://core.telegram.org/api

start the chatbot and name it

Send the following message to BotFather to create a new bot.

```
/newbot
```

Then choose a user name for the bot and it should end in 'bot' (e.g zijianBot or zijian_bot). The BotFather will provide you a link to find your Botand a token to access the HTTP API show as the following:

```
Done! Congratulations on your new bot. You will find it at t.me/zijianTestBot (Here is your

Use this token to access the HTTP API:
1505550933:AAFlZtYKApLkUR2DNRrS95gvrCxQxeoz9Bo
Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page: https://core.telegram.org/bots/api
```

For example,

- Link of my bot: t.me/zijianTestBot

- Token of my bot : 1505550933:AAFlZtYKApLkUR2DNRrS95gvrCxQxeoz9Bo

Now the chatbot can receive your message but can not response. You can send message to the chatbot and the logs can see the at the following website:

https://api.telegram.org/bot**YOUR_TOKEN_HERE**/getUpdates

and replace the text **YOUR_TOKEN_HERE** with your token.

Preparing the development environment

Now we can try to customize our own bot to make some simple response. You need to install the following a few modules.

Clone an empty repository from Github, or reuse the repository created in previous labs. Add a new file `requirements.txt` into your local folder with the following content:

```
telegram
configparser
redis
```

Note that you should use Python with version=3.7 or > 3.7. Type the following in the terminal/command prompt:

```
python -m pip install --upgrade pip
pip install -r requirements.txt
pip install python-telegram-bot==13.7
```

For security, we will using a config file to store the Token and the webhook link, first we make a config file `config.ini`.

```
[TELEGRAM]
ACCESS_TOKEN = YOUR_TOKEN_HERE
```

Simple echo chatbot

First we learn how to receive message from Telegram and echo to the message, we will use the following API,

telegram.ext.Updater : continuously fetches new updates from telegram and passes them on to the `Dispatcher` class

telegram.ext.Dispatcher: You can register different handlers in this class, it will sort the updates fetched by `Updater` according to the handlers you have registered.

telegram.ext.Handler: It contains subclass of handlers for different kind of updates (e.g. text,audio and so on)

telegram.ext.Fliters: It contain a number of filter to process the messages such as text, images and more.

You can find the detailed document of different API here.

Then we start to introduce how to produce a simple echo bot.

You can add a python file `chatbot.py` with the following source code to start a echo chatbot. Make sure you place the file together with `config.ini`

```
## chatbot.py
import telegram
from telegram.ext import Updater, MessageHandler, Filters
# The messageHandler is used for all message updates
import configparser
import logging


def main():
    # Load your token and create an Updater for your Bot
    config = configparser.ConfigParser()
    config.read('config.ini')
    updater = Updater(token=(config['TELEGRAM']['ACCESS_TOKEN']), use_context=True)
    dispatcher = updater.dispatcher
    # You can set this logging module, so you will know when and why things do not work as e
    logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
                        level=logging.INFO)

    # register a dispatcher to handle message: here we register an echo dispatcher
    echo_handler = MessageHandler(Filters.text & (~Filters.command), echo)
    dispatcher.add_handler(echo_handler)

    # To start the bot:
    updater.start_polling()
    updater.idle()


def echo(update, context):
    reply_message = update.message.text.upper()
    logging.info("Update: " + str(update))
    logging.info("context: " + str(context))
    context.bot.send_message(chat_id=update.effective_chat.id, text= reply_message)

if __name__ == '__main__':
    main()
```

run the following command on the terminal to start the chatbot.

```
python chatbot.py
```

or

```
py chatbot.py
```

or simply pressing the launch button from your IDE.

Then you can send the message to your bot in Telegram, and it can echo your messages.

You can also look at the log from your screen when you chat with your chatbot. After you have finished playing with your chatbot, press **Ctrl + C** to stop the program.

For more document about the telegram chatbot You can customize your chatbot using the document, and see more examples.

Push code

This is the end of Lab3. Please push your code to Github.

Reference:

1. python-telegram-bot official GitHub page: https://github.com/python-telegram-bot/python-telegram-bot

No write up for today's lab.