

## chatbot

- "[번역]" 단어가 들어가면 한글을 영문으로 번역해줌

### 1. 네이버 번역

In [1]:

```
import requests, json

def translate(naver_id, naver_secret, text, source="ko", target="en"):
    params = { "source": source, "target": target, "text": text }
    url = "https://openapi.naver.com/v1/papago/n2mt"
    headers = {
        "Content-Type": "application/json",
        "X-Naver-Client-Id": naver_id,
        "X-Naver-Client-Secret": naver_secret,
    }
    response = requests.post(url, json.dumps(params), headers=headers)
    return response.json()["message"]["result"]["translatedText"]
```

In [2]:

```
CLIENT_ID, CLIENT_SECRET = "6UJjfsjGSKlAJqme_T30", "6JIQkp0D06"
msg = translate(CLIENT_ID, CLIENT_SECRET, "파이썬은 재미있습니다.")
msg
```

Out[2]:

'Python is fun.'

### 2. 슬랙전송

- <https://api.slack.com/> (<https://api.slack.com/>) > incoming webhook 검색

In [3]:

```
def send_msg(webhook_url, msg, channel="#chatbot", username="슬랙봇"):
    payload = {
        "channel": channel,
        "username": username,
        "text": msg,
    }
    response = requests.post(webhook_url, data = json.dumps(payload))
    return response
```

In [4]:

```
WEBHOOK_URL = "https://hooks.slack.com/services/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"
msg = '안녕하세요.'
send_msg(WEBHOOK_URL, msg)
```

## 4. Create Flask App

In [7]:

```
!rm -rf chatbot
!mkdir -p chatbot/libs
!touch chatbot/chatbot.py
!touch chatbot/libs/naver.py
!touch chatbot/libs/slack.py
!tree chatbot
```

chatbot

```
├─ chatbot.py
└─ libs
    ├─ naver.py
    └─ slack.py
```

1 directory, 3 files

### 4.1 naver module

In [8]:

```
%%writefile chatbot/libs/naver.py
import requests, json

def translate(naver_id, naver_secret, text, source="ko", target="en"):
    params = { "source": source, "target": target, "text": text }
    url = "https://openapi.naver.com/v1/papago/n2mt"
    headers = {
        "Content-Type": "application/json",
        "X-Naver-Client-Id": naver_id,
        "X-Naver-Client-Secret": naver_secret,
    }
    response = requests.post(url, json.dumps(params), headers=headers)
    return response.json()["message"]["result"]["translatedText"]
```

Overwriting chatbot/libs/naver.py

In [13]:

```
# Test Translate
```

In [9]:

```
import chatbot.libs.naver as naver
```

In [10]:

```
msg = naver.translate(CLIENT_ID, CLIENT_SECRET, "파이썬은 재미있습니다.")  
msg
```

Out[10]:

'Python is fun.'

## 4.2 slack module

In [11]:

```
%%writefile chatbot/libs/slack.py  
import requests, json  
  
def send_msg(webhook_url, msg, channel="#chatbot", username="슬랙봇"):  
    payload = {"channel": channel, "username": username, "text": msg}  
    response = requests.post(webhook_url, data = json.dumps(payload))  
    return response
```

Overwriting chatbot/libs/slack.py

In [18]:

```
# Test Slack incoming webhook
```

In [12]:

```
import chatbot.libs.slack as slack
```

In [13]:

```
msg = '환영합니다.'  
slack.send_msg(WEBHOOK_URL, msg)
```

Out[13]:

<Response [200]>

## 4.4 flask route

In [3]:

```
%%writefile chatbot/chatbot.py
from flask import Flask, request, Response

import libs.naver as naver
import libs.slack as slack

app = Flask(__name__)

CLIENT_ID, CLIENT_SECRET = "6UJjfsjGSKlAJqme_T30", "6JIQkp0D06"
WEBHOOK_URL = "https://hooks.slack.com/services/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"

@app.route("/")
def index():
    return "server is running!"

@app.route("/bot", methods=['POST'])
def bot():

    username = request.form.get('user_name')
    token = request.form.get('token')
    text = request.form.get('text')

    print(username, token, text)
    text = text.replace('!bot ', '') # 트리거 워드 제거
    msg = naver.translate(CLIENT_ID, CLIENT_SECRET, text)
    slack.send_msg(WEBHOOK_URL, msg)

    return Response(), 200

app.run(debug=True, port=5000)
```

Overwriting chatbot/chatbot.py

In [ ]:

```
# !python chatbot/chatbot.py
```

## 5. flask config

- 설정 파일은 별도의 파일로 빼서 관리하면 관리가 편함

In [8]:

```
%%writefile chatbot/config.py

class Config(object):
    DEBUG = True
    TEMPLATES_AUTO_RELOAD = True
    CLIENT_ID = "6UJjfsjGSKlAJqme_T30"
    CLIENT_SECRET = "6JIQkp0D06"
    WEBHOOK_URL = "https://hooks.slack.com/services/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"
```

Overwriting chatbot/config.py

In [20]:

```
%%writefile chatbot/chatbot.py
from flask import Flask, request, Response

import libs.naver as naver
import libs.slack as slack

from config import Config

app = Flask(__name__)

app.config.from_object(Config) # app.run(debug=True)에 debug=True 설정을 안해도 됨

# print(Config.CLIENT_ID)
# CLIENT_ID = "6UJjfsjGSKlAJqme_T30"
# CLIENT_SECRET = "6JIQkp0D06"
# WEBHOOK_URL = "https://hooks.slack.com/services/T04LGD38EBW/B054CABMRM4/gGEJRp0jBSXDjC0YXqXhuh"

@app.route("/")
def index():
    return "server is running!"

@app.route("/bot", methods=['POST'])
def bot():

    username = request.form.get('user_name')
    token = request.form.get('token')
    text = request.form.get('text')

    print(username, token, text)

    text = text.replace('!bot ', '') # 트리거 워드 제거
    msg = naver.translate(Config.CLIENT_ID, Config.CLIENT_SECRET, text)
    slack.send_msg(Config.WEBHOOK_URL, msg)

    return Response(), 200

app.run(debug=True)
```

Overwriting chatbot/chatbot.py

- key 값은 코드에 공개가 되면 안됨
- key 값과 같은 민감한 값은 ini 확장자 파일이 저장해 놓고 가져오는 방법으로 사용

In [10]:

```
!pwd
```

/home/ubuntu/notebooks/flask

In [18]:

```
%%writefile data.ini
[var]
CLIENT_ID = 6UJjfsjGSKlAJqme_T30
CLIENT_SECRET = 6JIQkp0D06
WEBHOOK_URL = https://hooks.slack.com/services/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
```

Overwriting data.ini

In [16]:

```
!ls data.ini
```

data.ini

In [19]:

```
%%writefile chatbot/config.py

import configparser

config = configparser.ConfigParser()
config.read('../data.ini')
var = config["var"]

class Config(object):
    DEBUG = True
    TEMPLATES_AUTO_RELOAD = True
    CLIENT_ID = var["CLIENT_ID"]
    CLIENT_SECRET = var["CLIENT_SECRET"]
    WEBHOOK_URL = var["WEBHOOK_URL"]
```

Overwriting chatbot/config.py