

LINE chatbot のテスト環境構築

構築環境

Framework : Flask + line-sdk-python

Deploy : Heroku

手順1. LINEアカウントの作成と設定

参考 : <http://qiita.com/Kosuke-Szk/items/eea6457616b6180c82d3>

LINE ビジネスアカウントの作成

以下のURLから「DeveloperTrialを始める」を選択し、アカウントを作成する

<https://business.line.me/ja/services/bot>

アカウント設定

LINE@Managerの「アカウント一覧」から作成したアカウントを選択する.

「アカウント設定」-> 「Bot設定」から以下を設定して、保存する.

リクエスト設定

webhook送信 : 利用する

詳細設定

自動応答メッセージ : 利用しない



Bot設定 @ LINE developers

LINE developers への移動

LINE@Managerの「Bot設定」の中の "LINE Developersで設定する" をクリックし、LINE developersへ移動

アクセストークンの発行と保存

「Channel Access Token」の "ISSUE" をクリックし、アクセストークンを発行する。後の手順で、利用するので、Tokenをどこかに記録しておく

チャネルシークレットの保存

「Channel Secret」の "SHOW" をクリックし、チャネルシークレットを表示する。後の手順で、利用するので、チャネルシークレットをどこかに記録しておく

手順2. Flask アプリケーションの作成とHerokuへのデプロイ

(こっちはシンプル) 参考 : Flask を触ってみる + Heroku で動かす

<http://qiita.com/sqrtrxx/items/2ae41d5685e07c16eda5>

(こっちは詳細まで記載) 参考 : 続いてはPython (Flask) でHeroku!

<http://qiita.com/kounoike/items/6fc31fe051e5d6688f136>

pyenv + virtualenv による環境構築

環境構築

```
naist-wavenet125-221 GitHub/helloflask$ pyenv versions
system
* 2.7.11 (set by /Users/Yoshiharu/.pyenv/version)
  3.5.0
naist-wavenet125-221 GitHub/helloflask$ pyenv virtualenv 2.7.11
flask/2.7
pyenv-virtualenv: no slash allowed in virtualenv name.
naist-wavenet125-221 GitHub/helloflask$ pyenv virtualenv 2.7.11
flask_2.7
Collecting virtualenv
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
    100% |#####| 1.8MB 413kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-15.1.0
New python executable in
/Users/Yoshiharu/.pyenv/versions/2.7.11/envs/flask_2.7/bin/python2.7
Also creating executable in
/Users/Yoshiharu/.pyenv/versions/2.7.11/envs/flask_2.7/bin/python
Installing setuptools, pip, wheel...done.
Ignoring indexes: https://pypi.python.org/simple
Requirement already satisfied (use --upgrade to upgrade): setuptools
in
/Users/Yoshiharu/.pyenv/versions/2.7.11/envs/flask_2.7/lib/python2.7/site
packages
Requirement already satisfied (use --upgrade to upgrade): pip in
/Users/Yoshiharu/.pyenv/versions/2.7.11/envs/flask_2.7/lib/python2.7/site
packages
naist-wavenet125-221 GitHub/helloflask$ pyenv versions
system
* 2.7.11 (set by /Users/Yoshiharu/.pyenv/version)
  2.7.11/envs/flask_2.7
  3.5.0
  flask_2.7
```

環境の適用

```
naist-wavenet125-221 GitHub/helloflask$ pyenv local
2.7.11/envs/flask_2.7
(2.7.11/envs/flask_2.7) naist-wavenet125-221 GitHub/helloflask$ pyenv
version
2.7.11/envs/flask_2.7 (set by
/Users/Yoshiharu/Documents/GitHub/helloflask/.python-version)
```

Flaskの導入

pipによるinstall

```
GitHub/helloflask$ pip install flask
Collecting flask
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
    100% |#####| 92kB 959kB/s
----- 中略 -----
Successfully installed Jinja2-2.9.6 MarkupSafe-1.0 Werkzeug-0.12.2
click-6.7 flask-0.12.2 itsdangerous-0.24
```

installの確認

```
GitHub/helloflask$ pip list
click (6.7)
Flask (0.12.2)
itsdangerous (0.24)
Jinja2 (2.9.6)
MarkupSafe (1.0)
pip (9.0.1)
setuptools (36.0.1)
Werkzeug (0.12.2)
wheel (0.29.0)
```

Flask の Hello world 作成

以下のソースコードを作成し、作業ディレクトリに保存する

hello.py

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello world!'

if __name__ == '__main__':
    app.run()
```

hello.py のローカル実行

Terminal 上で hello.py を実行し、<http://127.0.0.1:5000/> にアクセスすると、Hello worldと表示される

```
GitHub/helloflask$ python hello.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Herokuアカウントの作成

HerokuのWebサイト（<https://signup.heroku.com/>）からアカウントを登録合わせて、CLIで heroku を操作するためのツールを導入しておく

```
GitHub/helloflask$ brew install heroku
```

Herokuで動かす

gunicornの導入

```
install
GitHub/helloflask$ pip install gunicorn
Collecting gunicorn
  Downloading gunicorn-19.7.1-py2.py3-none-any.whl (111kB)
    100% |#####| 112kB 1.1MB/s
Installing collected packages: gunicorn
Successfully installed gunicorn-19.7.1
```

heroku用のファイル準備

```
initialize_heroku_files
GitHub/helloflask$ pip freeze > requirements.txt
GitHub/helloflask$ echo python-2.7.11 > runtime.txt
GitHub/helloflask$ echo web: gunicorn hello:app --log-file=- >
Procfile
```

ファイル群を git 管理する

```
GitHub/helloflask$ git init
GitHub/helloflask$ echo .python-version > .gitignore
GitHub/helloflask$ git add .
GitHub/helloflask$ git commit -m "initial commit"
```

heroku のログインとデプロイ

```
login
GitHub/helloflask$ heroku login
Enter your Heroku credentials:
Email: xxxx@xxxx.jp
Password: *****
Logged in as xxxx@xxxx.jp
```

Heroku app の作成

ここで得られたURLを保存しておく。

手順3で、httpsのURLをLINE developersで登録する必要がある。

create

```
GitHub/helloflask$ heroku create
Creating app... done, 🍄 murmuring-garden-82706
https://murmuring-garden-82706.herokuapp.com/ |
https://git.heroku.com/murmuring-garden-82706.git
```

deploy(=git_push)

```
GitHub/helloflask$ git push heroku master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 673 bytes | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Python app detected
```

add_and_open_web_instance

```
GitHub/helloflask$ heroku scale web=1
GitHub/helloflask$ heroku open
```

手順3. LINE-sdk-python のテスト

line-sdk-python の導入

```
$ pip install line-bot-sdk
```

デモコードの作成

LINE bot sdk のドキュメント内の 「usage」 のソースコードを chatbot.py として保存する.

注) ソースコード内の "YOUR_CHANNEL_ACCESS_TOKEN", "YOUR_CHANNEL_SECRET" を手順1で保存したトークンに書き換えておく.

LINE bot sdk ドキュメント : <https://github.com/line/line-bot-sdk-python>

chatbot.py

```
from flask import Flask, request, abort

from linebot import (
    LineBotApi, WebhookHandler
```

```

)
from linebot.exceptions import (
    InvalidSignatureError
)
from linebot.models import (
    MessageEvent, TextMessage, TextSendMessage,
)

app = Flask(__name__)

line_bot_api = LineBotApi('YOUR_CHANNEL_ACCESS_TOKEN')
handler = WebhookHandler('YOUR_CHANNEL_SECRET')

@app.route("/callback", methods=['POST'])
def callback():
    # get X-Line-Signature header value
    signature = request.headers['X-Line-Signature']

    # get request body as text
    body = request.get_data(as_text=True)
    app.logger.info("Request body: " + body)

    # handle webhook body
    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        abort(400)

    return 'OK'

@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=event.message.text))

if __name__ == "__main__":
    app.run()

```

Heroku デプロイ用の準備

requirements.txt の更新

```
Github/helloflask$ pip freeze > requirements.txt
```

Procfile の更新

```
shell
```

```
before) web: gunicorn hello:app --log-file=-  
after ) web: gunicorn chatbot:app --log-file=-
```

更新内容を commit する

```
$ git add .  
$ git commit -m "update Procfile and requirements"
```

Heroku へのデプロイとテスト

デプロイ

```
deploy(=git_push)
```

```
GitHub/helloflask$ git push heroku master
```

LINE developers でアカウントを選択.

その後, 「Basic Information」-> 「Webhook URL」に Heroku app のURLを登録する.

注1) "https" で始まる方を登録する.

注2) URLの最後に"callback"をつけることを忘れない!! (例
"https://your_app_name.herokuapp.com/callback")

テスト

LINE developers の 「Basic Information」に表示されているQRコードを読み取り, チャットボットにアクセスして, テストする.

うまくいっていれば, オウム返しBotが完成しているはず!!!

