

Building a Customized Personal Assistant with Python

PyCon MY 2018

Iskandar Setiadi
Software Engineer at HDE, Inc.





Github

<https://github.com/freedomofkeima>

Blog

<https://freedomofkeima.com/blog/>

Speaker in:

- PyCon Japan 2017
- PyCon Indonesia 2018
- PyCon Japan 2018 (next month!)



Iskandar Setiadi

From Jakarta, Indonesia
Software Engineer at Japan

HDE, Inc. (<https://hde.co.jp/en/>)



Why Python?

- Easy to use, easy to learn
- Great community: nice & friendly!
- Swiss army knife: website development, data science, etc



Background



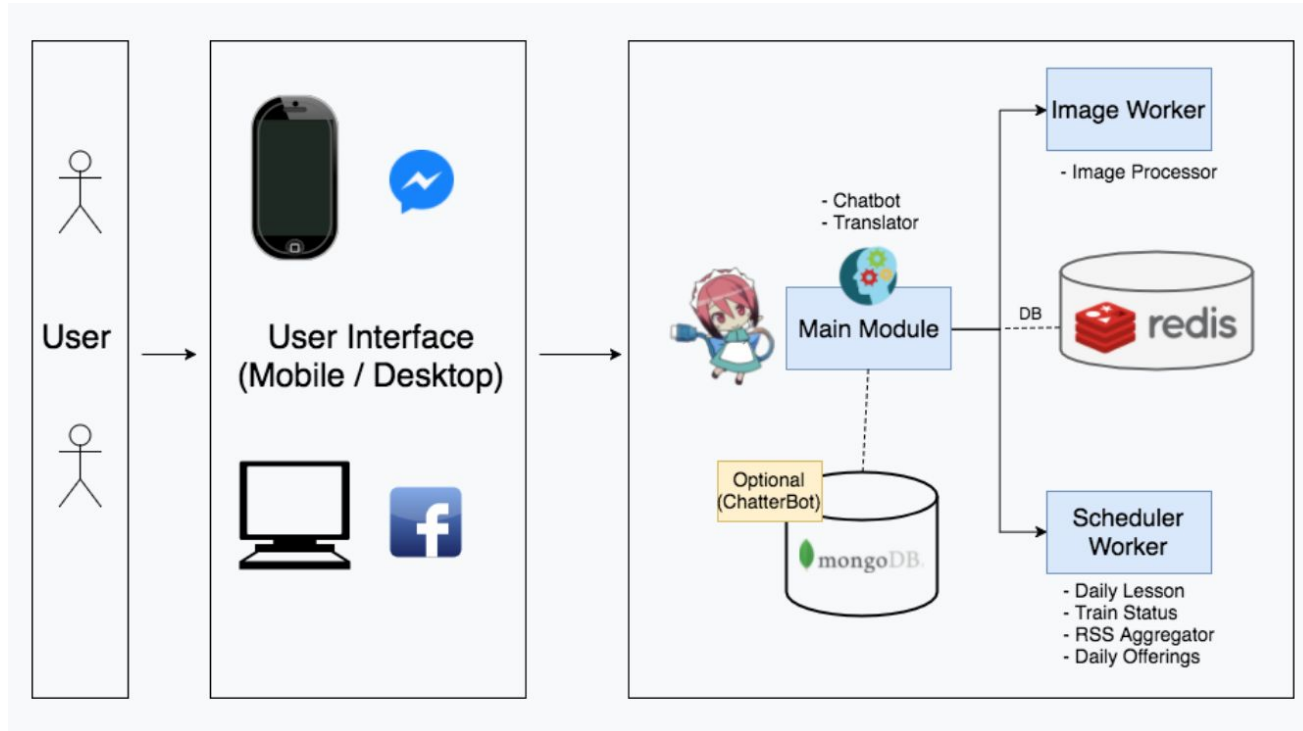


MAID-CHAN

Customized Personal Assistant



Overview



Why Facebook Messenger?

1. Multi-platforms

- Our bots can be accessed everywhere without developing own client-side interface.



iOS



Android



Facebook Canvas



Website

2. Huge number of users

- Facebook has more than 1 billion of users, which allows us to share our bots with our friends easily.



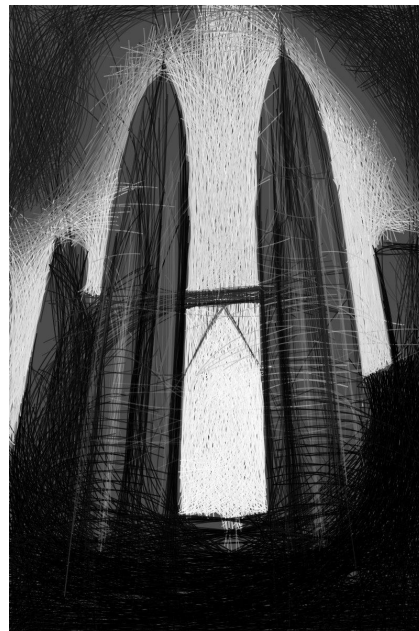
IMAGE FILTER WITH MACHINE LEARNING

Primitive: Reproducing images with geometric primitives

Github URL:

<https://github.com/fogleman/primitive>

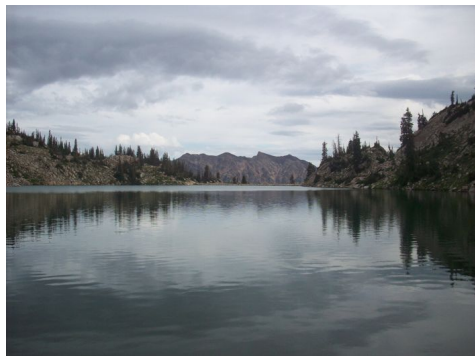
2000 bezier line filter →





Deep Photo Style Transfer

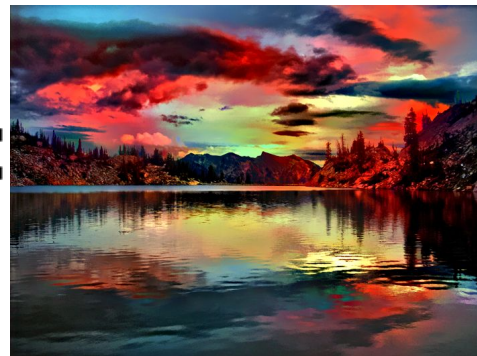
Github URL: <https://github.com/luanfujun/deep-photo-styletransfer>



INPUT



STYLE



OUTPUT



Machine Learning “Problems”

A lot of dependencies!

Caffee, Torch, CUDA, ~~Pikachu~~, GPU, ...

We cannot utilize these features to its fullest
in **mobile devices**.

Setup

This code is based on torch. It has been tested on Ubuntu 14.04 LTS.

Dependencies:

- [Torch](#) (with [matio-ffi](#) and [loadcaffe](#))
- [Matlab](#) or [Octave](#)

CUDA backend:

- [CUDA](#)
- [cudnn](#)

Download VGG-19:

```
sh models/download_models.sh
```

Compile `cuda_utils.cu` (Adjust `PREFIX` and `NVCC_PREFIX` in `makefile` for your machine):

```
make clean && make
```

Messenger bot to the rescue!



Configure Once, Execute Everywhere

```
def create_primitive_image(image_path, filename):
    for i in xrange(0, 3): # Generate 3 different images
        output_path = os.path.join(image_path, "output{}.png".format(i))
        args = [
            'primitive',
            '-i', # input file
            os.path.join(image_path, filename),
            '-o', # output file
            output_path,
            '-n', # number of iteration
            '175'
        ]
        # Execute
        p = subprocess.Popen(args, stdout=subprocess.PIPE, shell=False)
        (output, err) = p.communicate()
        if err:
            logging.error(err)
    return generate_gif_file_from_primitive(image_path, filename)
```

Generate a
Primitive image
as a GIF

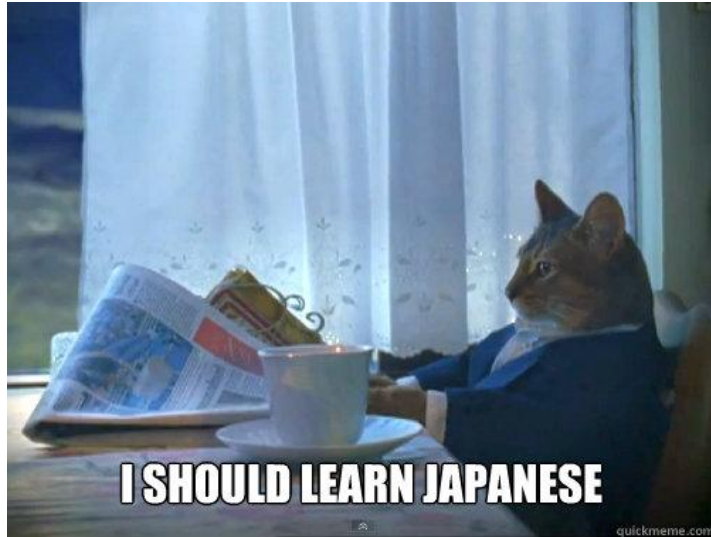
富嶽三十六景
神奈川沖
浪裏

江戶
葛飾
一筆

LEARN NEW LANGUAGE(S)



Living in Japan: Japanese Language





Step 1: Prepare your learning material

kanji	on	romaji on	kun	meaning	level	on...romaji kun
亜	ア アシア	a ashia	tsugu	Asia; rank next; come after; -ous;	1	ア アシア...つぐ
阿	ア オ	a o	omoneru	Africa; flatter; fawn upon; corner; nook; recess;	1	ア オ...おもねる
哀	アイ	ai	aware awaremu kanashii	pathetic; grief; sorrow; pathos; pity; sympathize;	1	アイ...あわれ あわれむ かなしい
愛	アイ	ai	itoshii	love; affection; favourite;	2	アイ...いとしい
葵	キ	ki	aoi	hollyhock;	1	キ...あおい
茜	セン	sen	akane	madder; red dye; Turkey red;	1	セン...あかね
悪	アク オ	aku o	warui waru ashi nikui aa izukuni izuk	bad; vice; rascal; false; evil; wrong;	3	アク オ...わるい わる- あしにくい ああ いくに いくんそ
握	アク	aku	nigiru	grip; hold; mould sushi; bribe;	1	アク...にぎる
渥	アク	aku	atsui uruou	kindness;	1	アク...あついうるおう
旭	キョク	kyoku	asahi	rising sun; morning sun;	1	キョク...あさひ

Raw data is downloaded from http://gakuran.com/files/japanese_ilpt1_lite.xls



Step 2: Load the data

Python csv module to help!

```
with open(filename, 'rb') as fobj:
    reader = csv.reader(fobj, delimiter=',', encoding='utf-8')
    ... # get random Kanji & Vocabulary based on level choice here
```



Step 3: Create a daily scheduler

```
while True:
    current_mt = int(time.time())
    metadata = redis_client.get_schedules()
    if metadata["next_mt"] < current_mt:
        metadata["kanji_n4"] = get_random_kanji(level=4)
        ... # Store any other daily information
        metadata["next_mt"] += 86400 # every 1 day
    redis_client.set_schedules(metadata)
```



Step 4.1: Subscribe for bot users

subscribe japanese



Thanks for subscribing to Maid-chan Japanese lessons ❤️
Which level of Kanji do you want to learn between N1-N4,
goshujin-sama? (e.g.: N3)

N2



Your information for Kanji level has been updated ❤️





Step 4.2: Enjoy the result

TUE 1:00PM

Welcome to Maid-chan's Daily Japanese lesson!

Today Kanji is 似

On: シ

Kun: にる ひる

Meaning: becoming; resemble; counterfeit; imitate; suitable

Today Vocabulary is かんたん

Kanji: 簡単

Meaning: simple



See you tomorrow in the same section! ❤️



AUTOMATIC TRANSLATION



Language Reference(s)





soimort/translate-shell

<https://github.com/soimort/translate-shell>

Translate Shell



build **passing** release **0.9.6.4** download **1.76 kB** [glitter](#) [join chat](#)

Translate Shell (formerly *Google Translate CLI*) is a command-line translator powered by [Google Translate](#) (default), [Bing Translator](#), [Yandex.Translate](#) and [Apertium](#). It gives you easy access to one of these translation engines in your terminal:

```
$ trans 'Saluton, Mondo!'
Saluton, Mondo!

Hello, World!

Translations of Saluton, Mondo!
[ Esperanto -> English ]
Saluton ,
  Hello,
Mondo !
  World!
```



Shlex - Simple Lexical Analysis

```
import shlex
try:
    elements = shlex.split(query.encode('utf-8'))
except:
    elements = [query.encode('utf-8')]

# Input   : Translate "I am going to Pycon" to 日本語
# Output  : ['Translate', 'I am going to Pycon', 'to', '日本語']
```




Utilize trans-shell + Google Translate

```
...
# Use trans online language detection
if not source_language_key:
    source_language_key = get_trans_language_prediction(source_text) # "trans -id {text}" handling
if not target_language_key:
    target_language_key = Constants.TRANSLATION_MAP[source_language_key]

args = ["trans", "-b", "{}:{}".format(source_language_key, target_language_key), source_text]
p = subprocess.Popen(args, stdout=subprocess.PIPE, shell=False)
(output, _) = p.communicate()
return output
```



Examples



私の名前はiskandarです

Translate my name is Iskandar



nama saya iskandar

Translate my name is Iskandar to bahasa



good Morning

Translate guten morgen



Examples



私は日本に住んでいる

Terjemahkan saya tinggal di Jepang



私はpyconに行くつもりです

Translate "I am going to Pycon" to 日本語



The weather today is nice

Translate 今日の天気はいいです

A digital illustration of a train at a station platform at night. The train is white and blue, with the number '20531' on its front. The destination sign above the train reads '新宿' (Shinjuku) in red and 'SHINJUKU' in green. The platform has a yellow tactile paving strip. The sky is dark blue with many white stars, and there are some lens flare effects. A semi-transparent black box with white text is overlaid on the center of the image.

PUBLIC TRANSPORT STATUS (TRAIN)

Tokyo Train Status Update

2:17PM

Tokyu Den-en-toshi Line Status Update:
Status: Preparing to resume operations
Reason: 市が尾駅で発生した人身事故の影...
Automatic translation: Shadow of injuries accident occurred at Ichigao station...

2:47pm

Tokyu Den-en-toshi Line Status Update:
Status: Delays
Reason: 12:51頃、市が尾駅で発生し...
Automatic translation: About 12:51, it occurred at Ichigao station...

4:46PM

Tokyu Den-en-toshi Line Status Update:
Status: Normal operations



PARSE



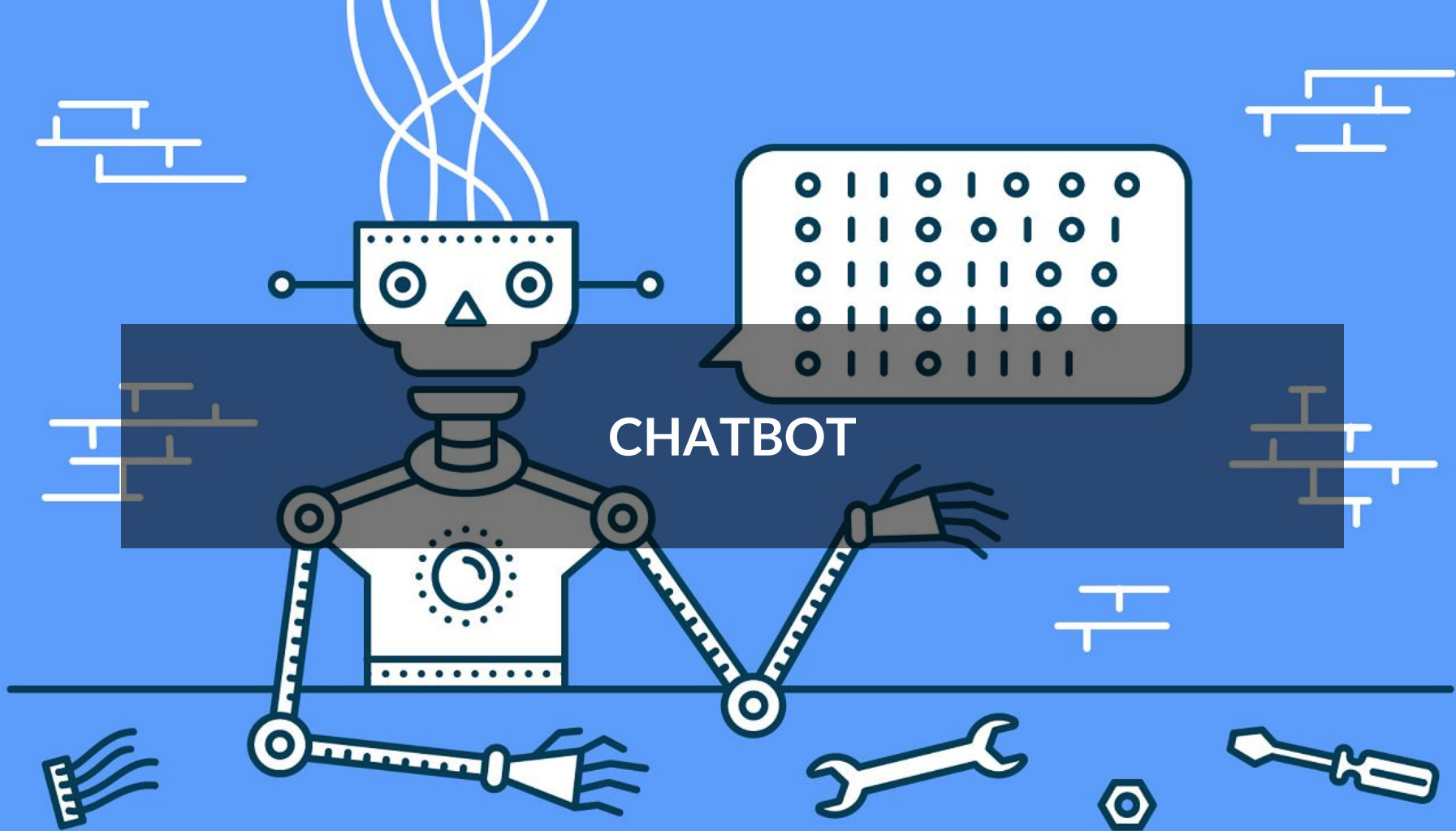


requests + BeautifulSoup + Google Translate

Re-use the feature that we have created before for automatic translation!

```
import requests
from bs4 import BeautifulSoup

# Retrieve page
r = requests.get("https://transit.yahoo.co.jp/traininfo/area/4/") # Yahoo Page
soup = BeautifulSoup(r.content, "html.parser")
soup_attrs = soup.select('div.trouble table tr td')
```





Showcase



Oyasumi~



OVER NINE THOUSAAAAAAAAND



My name is Maid-chan nanodesu~

Good night, Maid-chan!

May I know how old are you?

Wait, what is your name?



Chatterbot

```
INFO:root:Sender ID: 1242484182465009
INFO:root:Probable language is en
INFO:chatterbot.adapters:Recieved input statement: Wait, what is your name?
INFO:chatterbot.adapters:"Wait, what is your name?" is not a known statement
INFO:chatterbot.adapters:Using "Wait, what is your name?" as a close match to "W
hat is your name?"
INFO:chatterbot.adapters:Selecting response from 1 optimal responses.
INFO:chatterbot.response_selection:Selecting first response from list of 1 optio
ns.
INFO:chatterbot.adapters:Response selected. Using "My name is Maid-chan nanodesu
~"
INFO:chatterbot.adapters:<class 'chatterbot.logic.best_match.BestMatch'> selecte
d "My name is Maid-chan nanodesu~" as a response with a confidence of 0.86
INFO:chatterbot.adapters:<class 'chatterbot.logic.no_knowledge_adapter.NoKnowled
geAdapter'> selected "Wait, what is your name?" as a response with a confidence
of 0
```

Chatterbot

```
$ pip install chatterbot
```

```
from chatterbot import ChatBot

chatbot = ChatBot(
    'Ron Obvious',
    trainer='chatterbot.trainers.ChatterBotCorpusTrainer'
)

# Train based on the english corpus
chatbot.train("chatterbot.corpus.english")

# Get a response to an input statement
chatbot.get_response("Hello, how are you today?")
# Response: "I am doing well."
```



Custom Corpus

```
self.chatbot = ChatBot(
    'Maid-chan',
    trainer='chatterbot.trainers.ChatterBotCorpusTrainer', # Can also be trained with Twitter or Ubuntu
dialog corpus
    output_adapter="chatterbot.output.OutputFormatAdapter", # Output can be integrated with other framework
(MS Bot)
    output_format='text',
    storage_adapter="chatterbot.storage.MongoDatabaseAdapter" # Can be stored in JSON file, MongoDB, or
even SQL Alchemy
)

self.chatbot.train(
    "chatterbot.corpus.indonesia.conversations",
    "chatterbot.corpus.indonesia.greetings",
    "chatterbot.corpus.english.conversations",
    "chatterbot.corpus.english.greetings",
    "chatterbot.corpus.maidcorpus" # Train from a custom corpus
)
```



More Examples: Language-aware bot



Good morning to you too!

Selamat pagi, Maid-chan!



Good morning to you too!

Met pagi!



I am doing well.

So, how are you today?



RSS AGGREGATOR



Feedparser

```
$ pip install feedparser # https://github.com/kurtmckee/feedparser
```

```
import feedparser

response = feedparser.parse(url)
if 'bozo_exception' in response: # invalid response, not a valid RSS
    return False
for entry in response.get("entries", {}):
    title = entry.get("title")
    ...
```


A 3D rendered anime-style character with long dark hair, blue eyes, and a sailor-style outfit. She has a large red bow on her head and is wearing black lace wristbands. The background is a soft, glowing gradient with bokeh light effects.

BONUS: DAILY GREETINGS

Sky is the limit!

Make your bot more fun!

For example, you can add a feature which let your bot to send good morning / night messages →



Maid-chan

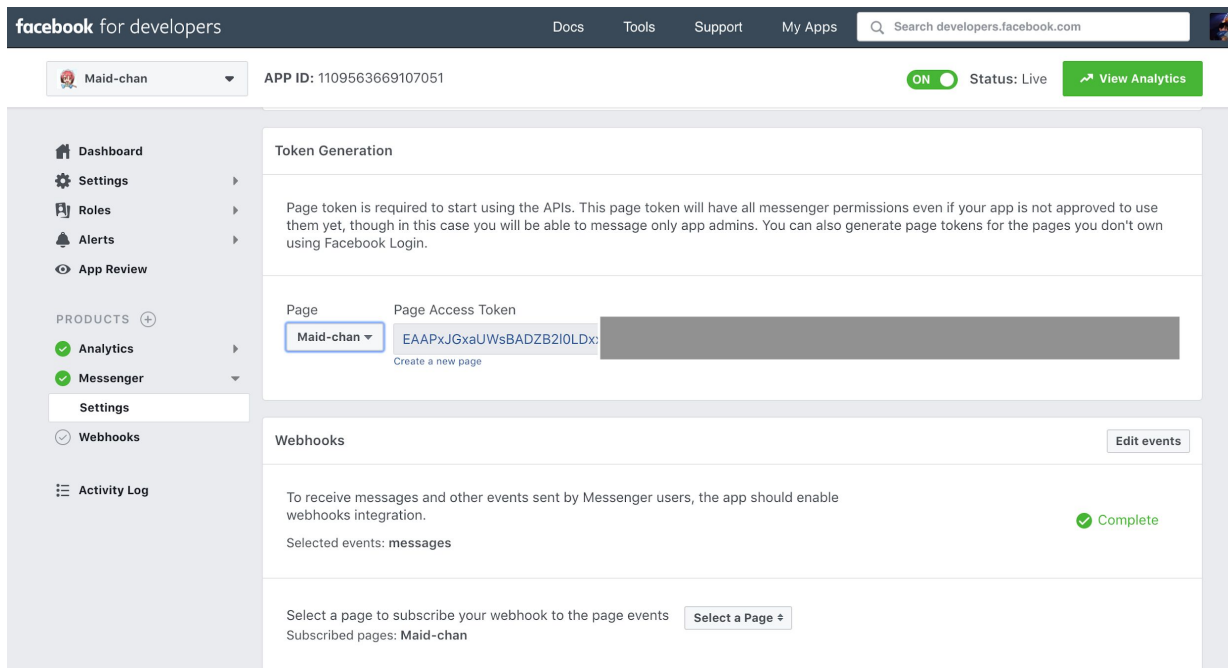
Oyaa~ Let's have a nice sleep, goshujin-sama



Maid-chan



Facebook Messenger Bot: Configure



The screenshot displays the Facebook for developers dashboard for an application named "Maid-chan" with APP ID: 1109563669107051. The status is "ON" and "Live". The left sidebar contains navigation options: Dashboard, Settings, Roles, Alerts, App Review, PRODUCTS (Analytics, Messenger), Settings (selected), Webhooks, and Activity Log. The main content area is titled "Token Generation" and includes a text block explaining that a page token is required to start using the APIs. Below this, there is a section for "Page Access Token" where the "Page" is set to "Maid-chan" and the token is "EAAPxJGxaUWsBADZB2i0LDx:". A "Webhooks" section below indicates that webhooks integration is "Complete" and lists "messages" as selected events. At the bottom, there is a "Select a page to subscribe your webhook to the page events" section with "Maid-chan" listed as a subscribed page.

URL: <https://developers.facebook.com/apps/>

Demo



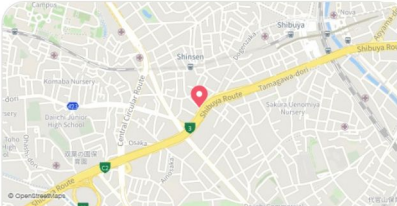
Caveats

- Facebook has 20 seconds timeout
- Facebook will keep sending user's message until a response is given
- Facebook has recently tightened some permissions control, e.g.: user's gender, user's locale, user's timezone, etc
- Without submission, we can invite friends as application testers
- Users are prone to make mistakes, ensure that you validate users' input

Infinite Possibilities: Find a restaurant

Optimist Prime Bot

Can you send me your location? 😊



Your Location

I've received location (35.655252,139.694218)

Zoom zoom zoom...

Okay, I've found 10 places:

Optimist Prime Bot

Options

- Search in Conversation
- Manage Messages
- Leave Feedback
- Notifications


Messenger Link

m.me/optimistPrimeBot

Infinite Possibilities: Find a restaurant

Optimist Prime Bot

Okay, I've found 10 places:



Gessekai - 4.5★ (2 reviews)
道玄坂2-26-5 (0.3 mi.)
Chinese

[View details](#)

Chōra
宇田川
Chine

Optimist Prime Bot

Options

- Search in Conversation
- Manage Messages
- Leave Feedback
- Notifications

Messenger Link

m.me/optimistPrimeBot

URL: <https://github.com/hungtraan/FacebookBot>

Infinite Possibilities

IoT switch (socket) allows us to connect our home devices with our phones. We can control our lamp, air conditioner, and so on over the internet!





Infinite Possibilities (General)

- Location-aware features: recommendation list, weather forecast, etc
- IoT integration features: air conditioner control, surveillance camera, etc
- Better NLP: Wit.ai (Facebook), IBM Watson, Google Cloud Speech, LUIS (Azure), Amazon Lex, etc



START BUILDING YOUR OWN BOT!

START BUILDING YOUR OWN BOT!



Github Project

<https://github.com/freedomofkeima/messenger-maid-chan>

freedomofkeima / messenger-maid-chan

Unwatch 9 Star 35 Fork 8

Code Issues 4 Pull requests 0 Projects 0 Wiki Insights Settings

Maid-chan feat Facebook Messenger bot to accompany personal daily life <http://messenger-maid-chan.rtfid.io> Edit

facebook-messenger-bot personal-assistant artificial-intelligence python facebook-messenger jarvis Manage topics

Presentation Slide: <https://freedomofkeima.com/pyconmy2018.pdf>

HDE, Inc. at Shibuya, Tokyo

Global Internship Program
<https://www.hde.co.jp/en/gip/>

PyCon MY 2018 **Gold** Sponsor

Visit our booth for more info!



Full-timer benefits: Get your CfP accepted and the company will handle the rest!





Thank you!



[freedomofkeima](#)
