Bot Messenger System

Version 0.1

Table of Contents

[I. Concept 3](#_Toc531686820)

[1. Web-hook server. 3](#_Toc531686821)

[2. Facebook integration. 3](#_Toc531686822)

[3. Messenger management system. 3](#_Toc531686823)

[II. Bot Messenger System diagram. 3](#_Toc531686824)

[III. Setup 4](#_Toc531686825)

[1. Web-hook server. 4](#_Toc531686826)

[2. Messenger management system. 5](#_Toc531686827)

[3. Facebook integration. 6](#_Toc531686828)

[a. Create application 6](#_Toc531686829)

[b. Add Facebook login 7](#_Toc531686830)

[c. Add Platform 7](#_Toc531686831)

[d. Add Web-hook 7](#_Toc531686832)

[e. Add Messenger 8](#_Toc531686833)

[IV. Test 9](#_Toc531686834)

# Concept

Bot Messenger System should have minimum three parts.

* Web-hook server.
* Facebook integration.
* Messenger management system.

## Web-hook server.

Web-hook server is a server responsible for receive events from Facebook, process and/or response the events.

## Facebook integration.

In order to integrate with Facebook echo system, we need to create a Facebook Application and setup necessary products binding to this app.

## Messenger management system.

This part takes responsibility to manage physical or logical connections between user Facebook assets and Web-hook server.

Example we can link or unlink a user Facebook page to one of our bot messengers.

# Bot Messenger System diagram.

Facebook

Web hook

Bot Messenger management system

Connections database

Bot#1

Bot#2

Bot#...

Bot#n

Subscribe/unsubscribe app

Link/unlink user asset and Bot

Facebook events

Finding connection of user asset and Bot

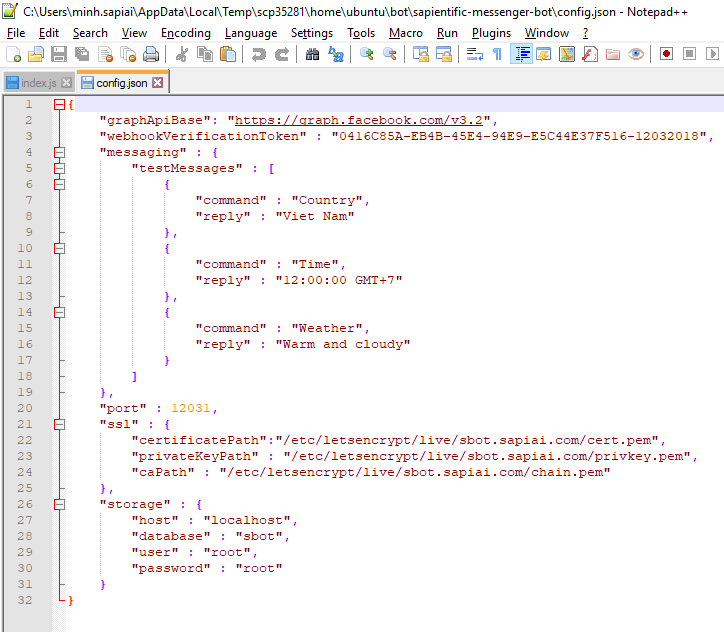
# Setup

## Web-hook server.

In order to receive events from Facebook, the Web-hook server must have a SSL callback URL.

So, before run the Web-hook application, it must prepare a trusted SSL certificates include certificate file, private key file and CA file.

Current web-hook application (sapientific-messenger-bot) running in Node.js. So, Node.js is also required on the host.

  
sapientific-messenger-bot configuration file

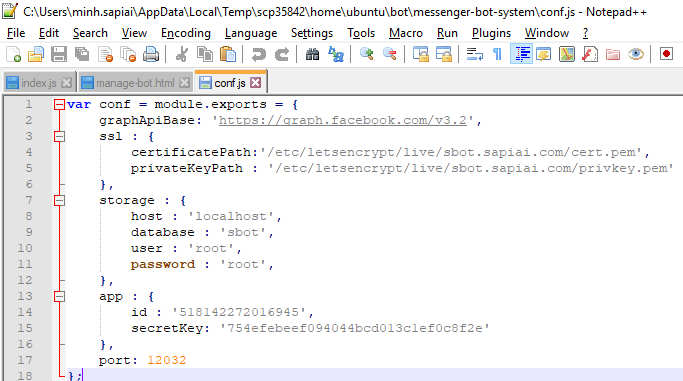
* graphApiBase: URI base of Facebook APIs used in this app. Note that, the API version should match with Facebook app that will be created later. Be careful to change version of API, it may cause of the application does not work properly.
* webhookVerificationToken: use to verify web-hook callback URL. Only used at the first time that web-hook added to on Facebook App (see [Facebook integration](#_Add_Web-hook) setup step for more details).
* messaging: auto reply message for testing purpose only.
* port: listening port of this application.
* ssl: ssl encryption information, require certificate file, private key file and CA file.  
  This filed is optional, if it is omit then the server will start listening on HTTP instead of HTTPS.
* storage: MySQL connection information

## Messenger management system.

Messenger management server expose web user interface on Facebook App’s domain, so this server must be listening on a SSL connection too.

Besides that, this server should contain APIs that responsible for calling Facebook API to manage subscriptions on Facebook and connections on local database.

Currently, messenger-bot-system is written in Node.js does all above operations.

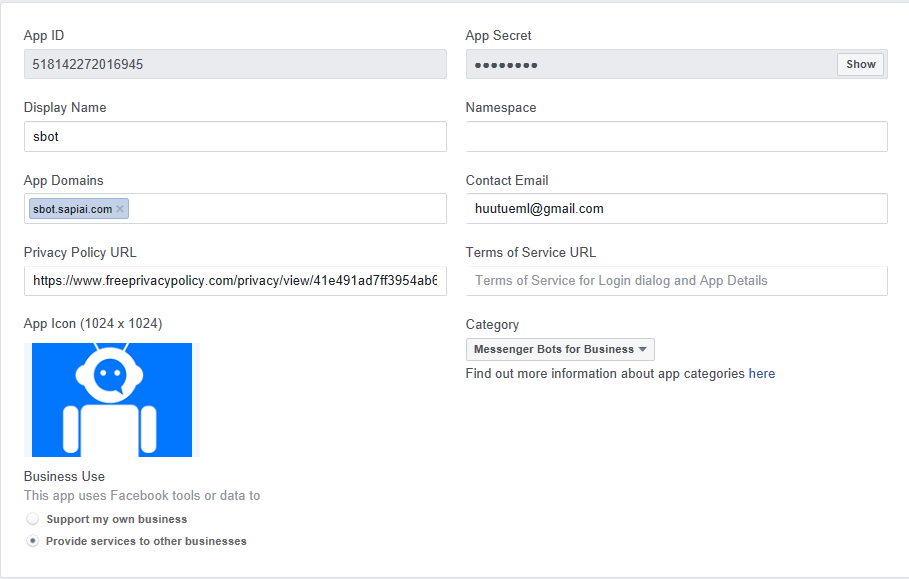
  
messenger-bot-system’s configuration file

* graphApiBase: URI base of Facebook APIs used in this app. Note that, the API version should match with Facebook app that will be created later. Be careful to change version of API, it may cause of the application does not work properly.
* ssl: ssl encryption information, currently require only certificate file and private key file.  
  This filed is optional, if it is omit then the server will start listening on HTTP instead of HTTPS.
* storage: MySQL connection information.
* app: Facebook app’s information.  
   id: Facebook App ID.  
   secretKey: Facebook App Secret.  
  They should be match with information of Facebook app that will be created later.
* port: listening port of this application.

## Facebook integration.

### Create application

Login to [Facebook developer](https://developers.facebook.com) and create an application as bellow.



* App ID: Facebook App ID

This filed is used in messenger-bot-system and sapientific-messenger-bot.

* App Secret: Facebook App Secret.

This filed is used in messenger-bot-system and sapientific-messenger-bot.

* App Domains: domain of all products such as website, Facebook Login, Web-hook.

Currently, messenger-bot-system contains website and Facebook Login while sapientific-messenger-bot contain web-hook. All these domains should be listed in App domains.

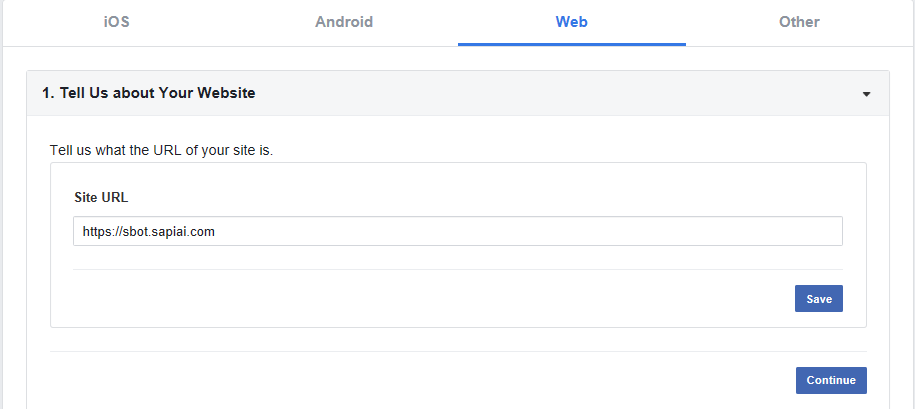
* Privacy Policy URL: this is an URL containing application privacy policy. This acts mainly as declaration of application privacy policy to ensure that any data collect from user should be reviewed.

Note that, all these information should be correct to not only ensure the system works properly but also to make the approval progress run smoothly(when the Facebook App change to public stage).

### Add Facebook login

Click Add Products and click Setup on Facebook Login‘s widget.

Fill the website here, no matter what it is but the website domain should be listed on App Domains above.



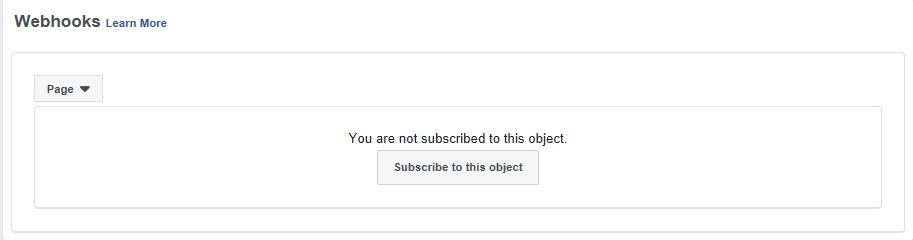
Just click Save and ignore other steps.

### Add Platform

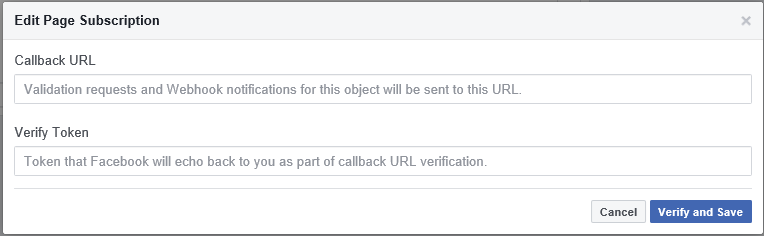
Currently, we only allow user login on browser, so we don’t need to add other platform. In future if we allow user login on iOS or Android, we should add more platform here.

### Add Web-hook

Click Add Products and click Setup on Webhooks’s widget



Select ‘Page’ and click Subscribe…



Callback URL: Web-hook listening URL that was run by sapientific-messenger-bot.

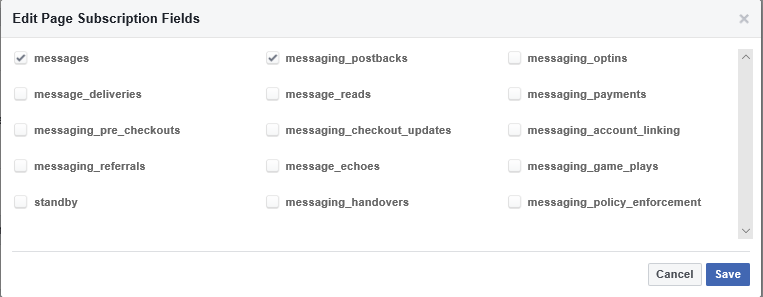
Verify Token: It is ‘webhookVerificationToken’ field of sapientific-messenger-bot configuration file.

After fill all these filed, click ‘Verify and Save’, if Facebook verify Web-hook success, the dialog is automatic close.

### Add Messenger

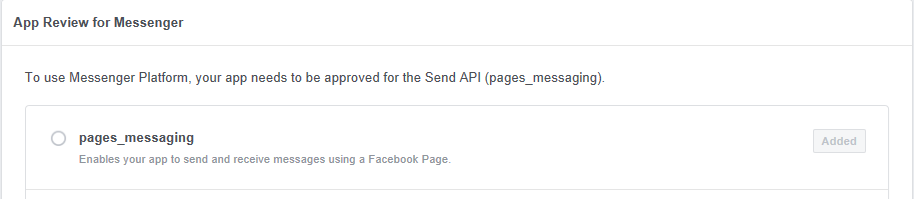
Click Add Products and click Setup on Messenger’s widget.

Click Edit Events in Webhooks’s widget.



Select event like above picture and click ‘Save’.

In App Review for Messenger’s widget. Add ‘page\_messaging’ to Submission.



This necessary when Facebook App submit for review.

Now, all settings are done. We can test the system.

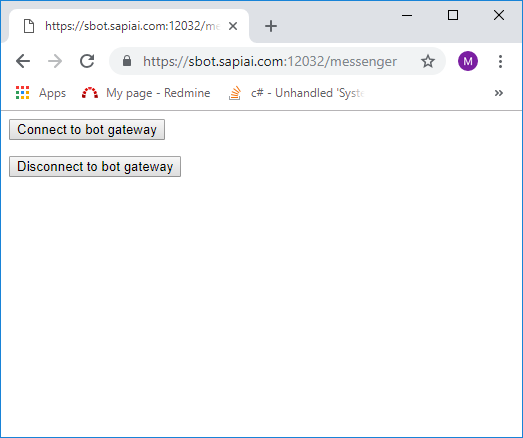
# Test

Assume that we deploy the whole system on domain sbot.sapiai.com and the configuration files are same as above.

In order to test the system do following steps.

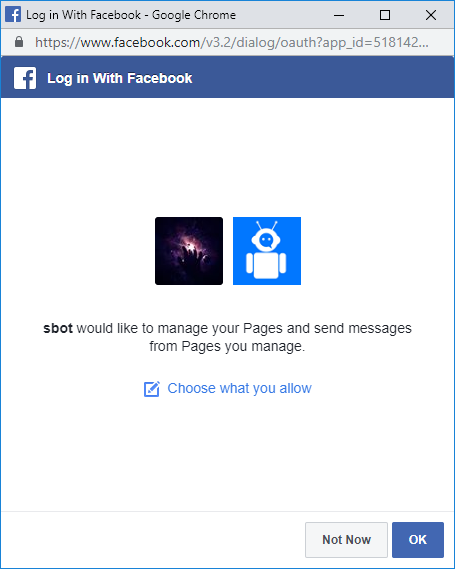
1. Add a Facebook account as Role tester for the app.
2. Add a test page for the Facebook account.
3. Open browser and access to this link.

<https://sbot.sapiai.com:12032/messenger>

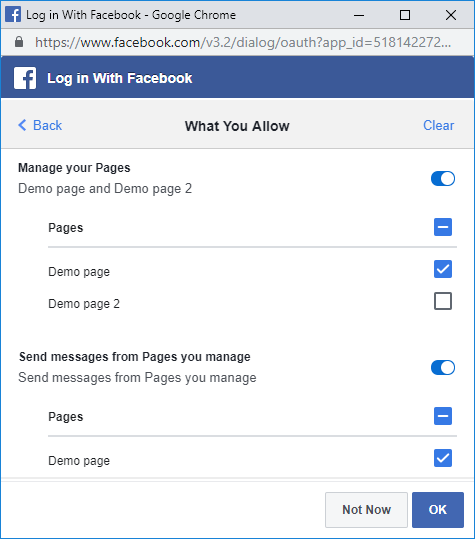


Click ‘Connect to bot gateway’, then login the Facebook account.

After login success, the dialog will ask for permissions.

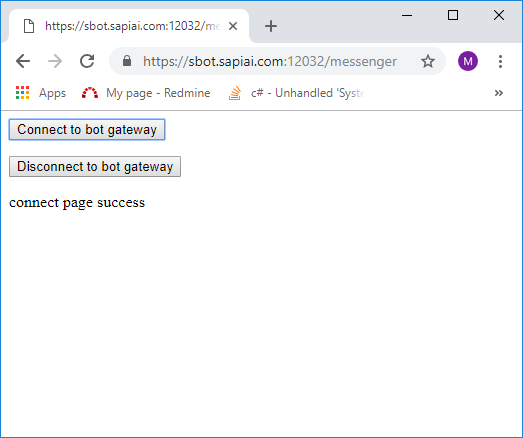


Click ‘Choose what you allow’.



Select only a Page that you want to connect to the Web-hook(sapientific-messenger-bot) then click OK.

If the connection is created successfully. You will see result like bellow.



1. Login to another Facebook account and send a message to the page.

The page should reply like bellow picture.

