
Build a slackbot

Use a slack bot to chat with your Watson Assistant Solutions tenant.

About this task

Create a slack bot that routes requests and responses in text to and from Watson Assistant Solutions.

Before you begin

Complete these steps to set up your environment:

1. Create a custom slack bot user

Create a new bot user for your slack account. Copy the username and API token for later use. For instructions, see the Custom bot users section of [Bot Users](#).

2. Set up your python environment

Complete these steps:

1. Install python version 3.6 from the [python.org website](#).
2. Install the python packages `python-dotenv`, `slackclient`, `sys`, and `requests`. For example:

```
pip install python-dotenv slackclient sys requests
```

Procedure

Complete these steps to build a slack bot locally:

1. Clone the [Watson Assistant Solutions slack bot](#).
2. Find the `../simple_WA_slackbot/.env.sample` and rename it `.env`. The file must be in the root folder. **Tip:** Keep any credential configuration files private.
3. Edit the file to include the following:
 - logging level
 - slack bot user credentials
 - Watson Assistant Solution credentials
 - Fallback phrase

Example:

```
# Log Level (DEBUG, INFO, WARNING, ERROR)
LOG_LEVEL="WARNING"
```

```
# Slack Credentials
SLACK_API_TOKEN="Place_your_slack_bot_API_token_here"
BOT_ID=""Place_your_bot_user_name_here""

# WA Credentials
WA_URL="https://watson-personal-assistant-toolkit.mybluemix.net"
WA_SKILLSET="place_your_skillset_name_here"
WA_API_KEY="place_your_skillset_name_here"
WA_LANGUAGE="en-US"
WA_DEVICE_TYPE="slackbot"

# Fallback Phrases, Comma separated. OPTIONAL (Leave as empty string if
undesired)
FALLBACK_RESPONSES="I didn't quite catch that, I don't understand"
```

4. Start your slack bot. Enter:

```
python3 bot.py
```

5. Converse with your Watson Assistant Solutions tenant using slack. Either:

- Send a direct message to your bot on slack.
- Invite your slack bot to a slack channel. Enter `@botname _question_` to chat.

Hosting your slack bot on IBM Cloud

Complete these steps:

1. Install the [IBM Cloud CLI](#).
2. Log in to IBM Cloud from the command line. Enter:

```
bx login --sso
```

3. Push your slack bot application to IBM Cloud. Assign a name to the application and enter:

```
bx cf push $PLACE_YOUR_APP_NAME_HERE --no-route true --health-check-type process
```

1. Add VCAP variables. For instructions, see the various options [here] (https://console.ng.bluemix.net/docs/manageapps/depapps.html#ud_env) and [here](#).
2. Restage your application through the IBM Cloud UI or from the command-line. For example:

```
bx cf restage $YOUR_APP_NAME
```

Testing your slack bot application

To run all unit tests, enter:

```
python3 -m unittest discover
```

To run a specific unit test, enter:

```
python3 -m unittest test.test_env_file
```

General Logs

All chat bot logs are in the /slackbot.log file. The file mostly consists of requests, responses, and slack logging.

Logging unrecognized intents

You can log all responses that return a fallback response or no response from your assistant. The responses are written to the /fallback_responses.csv file.

To enable the logging of unrecognized intents, add the fallback responses in `FALLBACK_RESPONSES` variable in the .env file update please ensure that in your local .env or variables (local) or in your VCAP environment variables (IBM Cloud) you add comma-separated responses that come from your fallback skill.

For example:

```
# Fallback Phrases, Comma separated. OPTIONAL (Leave as empty string if undesired)
FALLBACK_RESPONSES="I didn't quite catch that, I don't understand"
```