

---

## title: Creating a slack bot weight: 65

Use the sample chat bot that is provided with Watson Assistant Solutions to chat with your assistant.

### Setting up your environment

To create an API token for your slack bot, complete these steps:

1. Go to [https://YOUR\\_SLACK.slack.com/](https://YOUR_SLACK.slack.com/)
2. In the Search App Directory field, enter Bots.
3. Click Bots.
4. Click Add Configuration.
5. Assign your bot a unique user name.
6. Save the API token and the user name for later use.

You will use the user name and API token when you set up your environment.

To set up a python environment, install python version 3.6.2 or later from the [python.org website](https://python.org).

### Building a slack bot

Complete these steps to build a slack bot locally:

1. Clone the [Watson Assistant Solutions slack bot](#).
2. Copy the sample `.env` file. On Linux or Mac OS, enter: `cp .env.sample .env`. On Windows systems, enter: `xcopy .env.sample .env`. The file must be in the root folder. **Tip:** Keep any credential configuration files private.
3. Edit the `.env` file. For example:

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

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

# Waton Assistant Soltuions Credentials
WA_URL="https://watson-personal-assistant-toolkit.mybluemix.net"
WA_SKILLSET="place_your_assistant_skillset_name_here"
WA_API_KEY="place_your_watson_assistant_solutions_api_key_here"
WA_LANGUAGE="en-US"
WA_DEVICE_TYPE="slackbot"

# Bot Configuration - Number of characters before card data is made into a
JSON snippet
MAX_CARD_CHARACTERS=1500
```

```
# (OPTIONAL) Configurations for Dashbot analytic services
ANALYTICS_ENABLED="FALSE"
ANALYTICS_API_KEY=""
ANALYTICS_INPUT_URL="https://tracker.dashbot.io/track?
platform=slack&v=9.8.0-rest&type=incoming&apiKey="
ANALYTICS_RESPONSE_URL="https://tracker.dashbot.io/track?
platform=slack&v=9.8.0-rest&type=outgoing&apiKey="

# Max Message Pointer Cache Size
MAX_MESSAGE_CACHE=1000
```

4. Install the required python libraries. For example:

```
pip3 install -r requirements.txt
```

5. Test that your slack bot is working. To run all unit tests, enter:

```
python3 -m unittest discover
```

To run a specific unit test, enter:

```
python3 -m unittest test.<test name>
```

Where values for `_test name_` includes `test_WA_configuration`, `test_slack_configuration`, `test_env_file`, `test_context_file`, and `settings`.

6. Start your slack bot. Enter:

```
python3 bot.py
```

7. Chat with your slack bot. Either:

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

## Hosting your slack bot on IBM Cloud

After you test that your slack bot is running locally, push the bot to IBM Cloud for others to use.

1. Install the [IBM Cloud CLI](#).

2. Log in to IBM Cloud.
3. Go to your slack bot directory and push your slack bot application to IBM Cloud. Assign a name to the application. Enter:

```
cf push $YOUR_APP_NAME_HERE --no-route true --health-check-type process -m 256M
```

Or, depending on your CLI tools, enter:

```
bx cf push $YOUR_APP_NAME_HERE --no-route true --health-check-type process -m 256M
```

4. Verify that you can chat with your bot. Send it a direct message to your bot on slack.

## Adding context

You can simulate context variables in your slack bot conversation. Provide the context information in the `/context.json` file. The file must be in valid JSON format. The context information is sent in the request to the Watson Assistant Solutions core routing component.

Use this mechanism if you are testing a skill that requires location information.

## Viewing logs

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

## Viewing analytics

Your slack bot has built in hooks for enabling analytics. For example, you can enable analytics for dashbot.io by setting `ANALYTICS_ENABLED=TRUE` and adding in your dashbot.io API key to the `.env` file.

### What to do next?

Learn how to [integrate with IFTTT](#).