MARK5826 - MovieBot

Tutorial Week 4

Windows 10 Version

Table of Contents

Intro

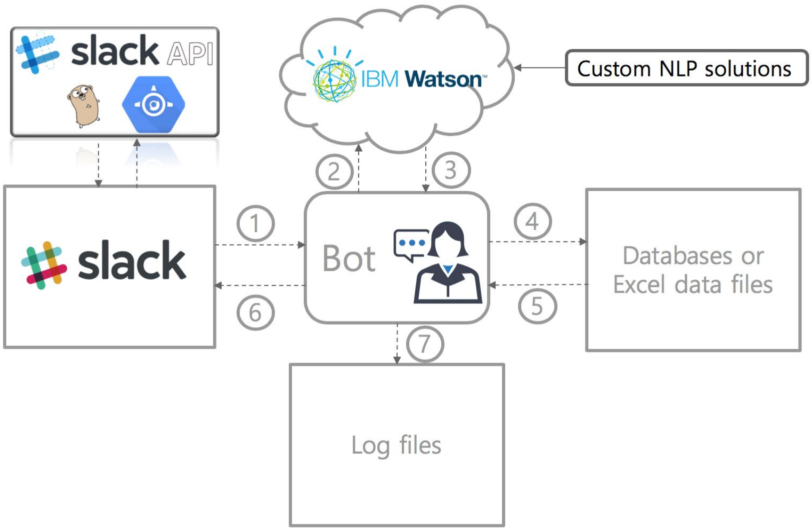
In this tutorial we will deploy a bot which has capability to give movies information and gives movie recommendation based on similar genre. The objective of this tutorial :

- Understand Chatbot Custom Integration
- Create Slack Bot APP
- Create Watson skill from json file
- Preparing dataset
- Deploy and Run Bot Backend

Tools we need:

- Browser: Mozilla Firefox, Google Chrome, Microsoft Edge (Included with Windows)
- Text Editor: Notepad (Included with Windows) , Sublime or Visual Studio Code
- Anaconda Prompt

The Movie bot framework used here is a closed domain chatbot. The entire framework design is shown below.



Step 1 (User asks question):

Users can interact with Kelly via Slack. Once the user post a question via the interface, the question is passed to the backend system for analysis

Step 2 (NLP processing):

All the natural language processing happens in step 2.

Step 3 (Return the NLP results):

After the NLP processing is completed, we have three outputs from it

1. Intents - What the user is trying to ask or query?
2. Entities - What is the exact field or column they are looking for?
3. Dialog/Interaction - Provide the appropriate request/response for the user question.

Step 4 and 5(Query the data):

Currently, the data resides in a excel file. However, you can add multiple databases/excel files if needed, to access different sources. Based on the results from step 3, the appropriate database/excel file is queried and the results are returned.

Step 6 (Post the result to user):

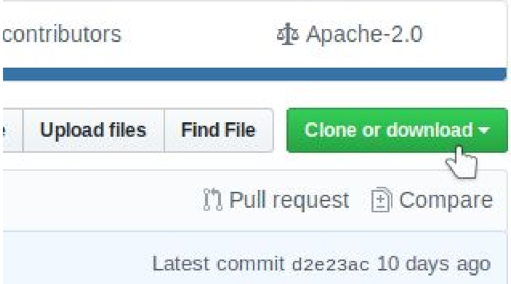
The results obtained from the backend is posted to user via Slack

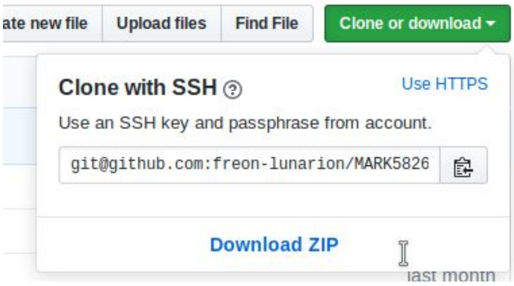
Step 7 (Log maintenance):

The interactions between the users are logged and stored in a flatfile format in a log file. Also, if the bot is not able to identify the user questions it will add those questions to a followup file.

Act 1 - Get the project source code

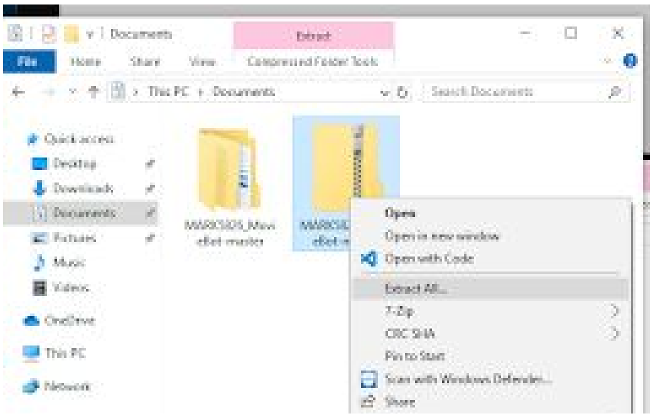
1. Go to https://github.com/freon-lunarion/MARK5826_MovieBot
2. Click "**Clone or download**" Button, select on "**Download ZIP**"

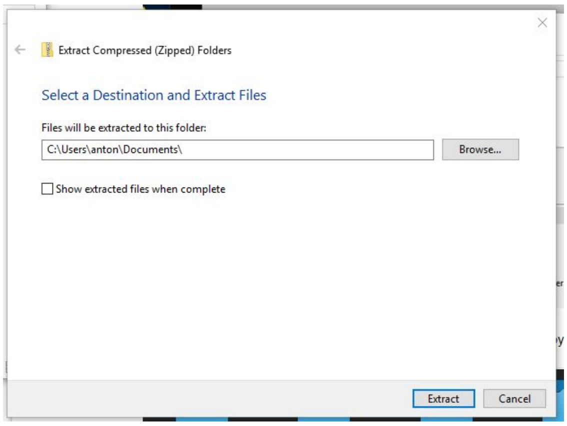




git@github.com:freon-lunarion/MARK5826

3. Extract the **MARK5826_MovieBot-master.zip** file. Note : you can move or rename the extration folder to other place or name.

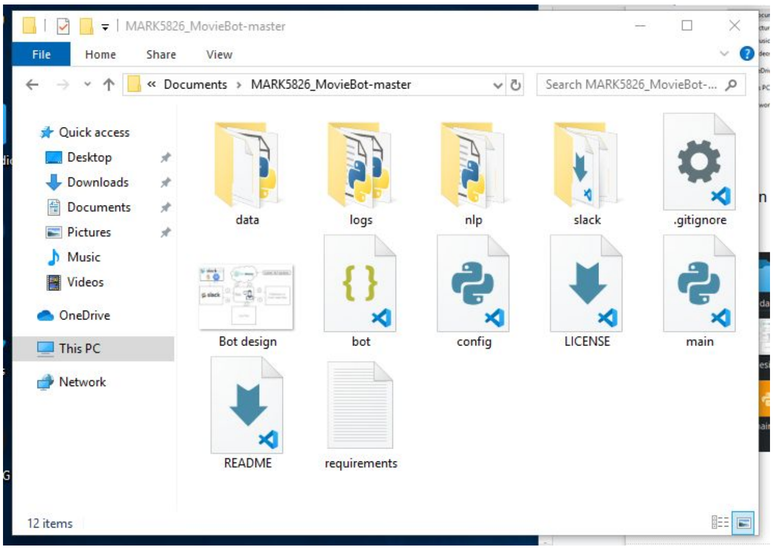



Extract Compressed (Zipped) Folders
Select a Destination and Extract Files
Browse...
Extract
Cancel

Files will be extracted to this folder:
C:\Users\anton\Documents\

□ Show extracted files when complete

4. Open the **project** folder, you will have 4 folders, 2 python files(.py), 1 text file (.txt) and 1 json file



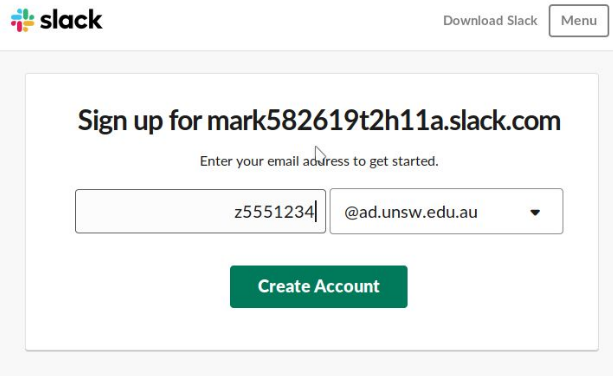
Act 2 - Create a slack app

Step1 - Login to your workspace

Open link below to create slack account (choose one):

- Thursday 11am:
  - https://join.slack.com/t/**mark582619t2h11a**/signup
  - https://join.slack.com/t/**mark582619t2h11b**/signup
  - https://join.slack.com/t/**mark582619t2h11c**/signup
- Thursday 1pm:
  - https://join.slack.com/t/**mark582619t2h13a**/signup
  - https://join.slack.com/t/**mark582619t2h13b**/signup
  - https://join.slack.com/t/**mark582619t2h13c**/signup
- Thursday 4pm:
  - https://join.slack.com/t/**mark582619t2h16a**/signup
  - https://join.slack.com/t/**mark582619t2h16b**/signup
  - https://join.slack.com/t/**mark582619t2h16c**/signup
- Tuesday 9am:
  - https://join.slack.com/t/**mark582619t2t09a**/signup
  - https://join.slack.com/t/**mark582619t2t09b**/signup
  - https://join.slack.com/t/**mark582619t2t09c**/signup
- Tuesday 11am:
  - https://join.slack.com/t/**mark582619t2t11a**/signup
  - https://join.slack.com/t/**mark582619t2t11b**/signup
  - https://join.slack.com/t/**mark582619t2t11c**/signup
- Tuesday 2pm:
  - https://join.slack.com/t/**mark582619t2t14a**/signup
  - https://join.slack.com/t/**mark582619t2t14b**/signup
  - https://join.slack.com/t/**mark582619t2t14c**/signup

Use your UNSW email to join slack workspace

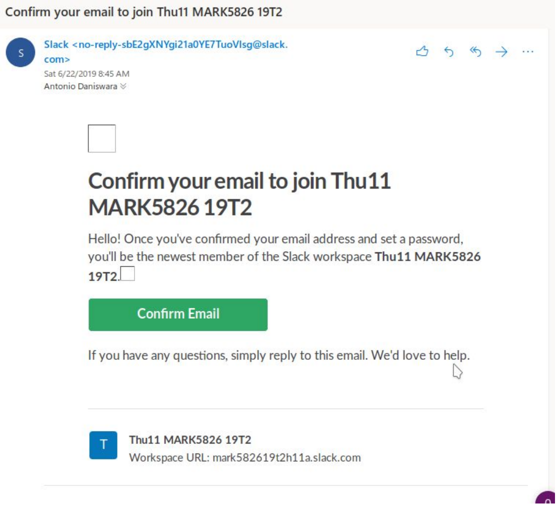


Sign up for mark582619t2h11a.slack.com

Enter your email address to get started.

z5551234 @ad.unsw.edu.au

Open your email and look the confirmation email from Slack. Click the Confirm Email button



Confirm your email to join Thu11 MARK5826 19T2

Slack <no-reply-sbE2gXNYgi21a0YE7TuoVlsg@slack.com>
Sat 6/22/2019 8:45 AM
Antonio Daniswara

□

Confirm your email to join Thu11 MARK5826 19T2

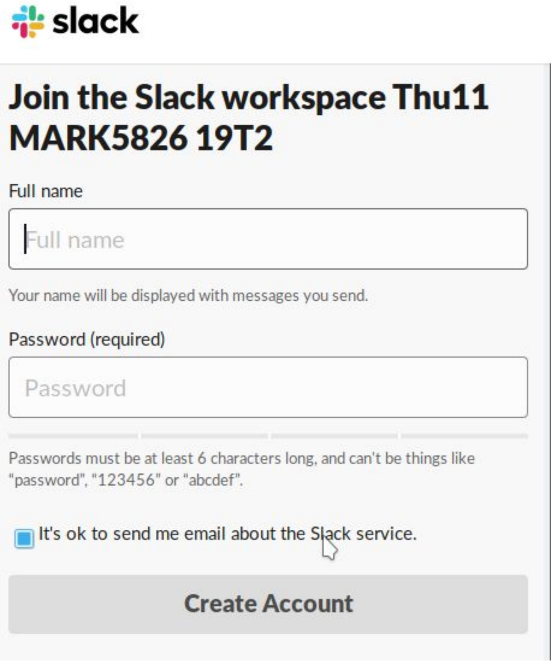
Hello! Once you've confirmed your email address and set a password, you'll be the newest member of the Slack workspace **Thu11 MARK5826 19T2**.□

Confirm Email

If you have any questions, simply reply to this email. We'd love to help.

T Thu11 MARK5826 19T2
Workspace URL: mark582619t2h11a.slack.com

Type your name and password. Click Create Account



Join the Slack workspace Thu11 MARK5826 19T2

Full name

Your name will be displayed with messages you send.

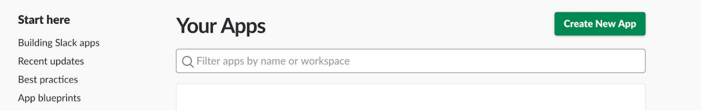
Password (required)

Passwords must be at least 6 characters long, and can't be things like "password", "123456" or "abcdef".

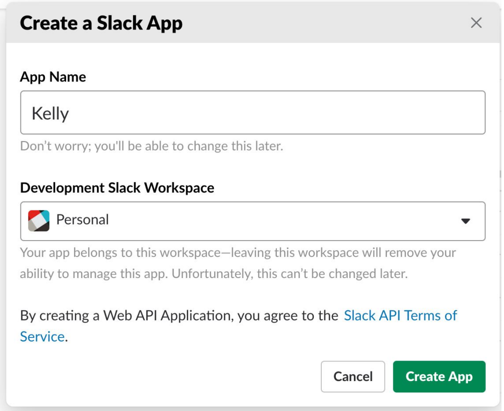
☑ It's ok to send me email about the Slack service.

Step2 - Build a slack app

Visit https://api.slack.com/apps to create a slack app.



Click on "Create New App" and then you will be redirected to the prompt below.



App Name: Kelly

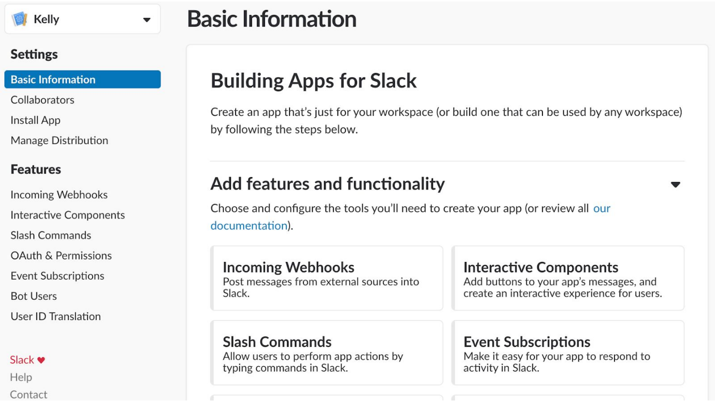
Don't worry; you'll be able to change this later.

Development Slack Workspace: Personal

Your app belongs to this workspace—leaving this workspace will remove your ability to manage this app. Unfortunately, this can't be changed later.

By creating a Web API Application, you agree to the Slack API Terms of Service.

Provide the App name with **[your zID]_app** (i.e. ***z5551234_app***) and the slack workspace you would like to install the app. After that, you can click on create app and you will be redirected to the app page.



Building Apps for Slack

Create an app that's just for your workspace (or build one that can be used by any workspace) by following the steps below.

Add features and functionality

Choose and configure the tools you'll need to create your app (or review all our documentation).

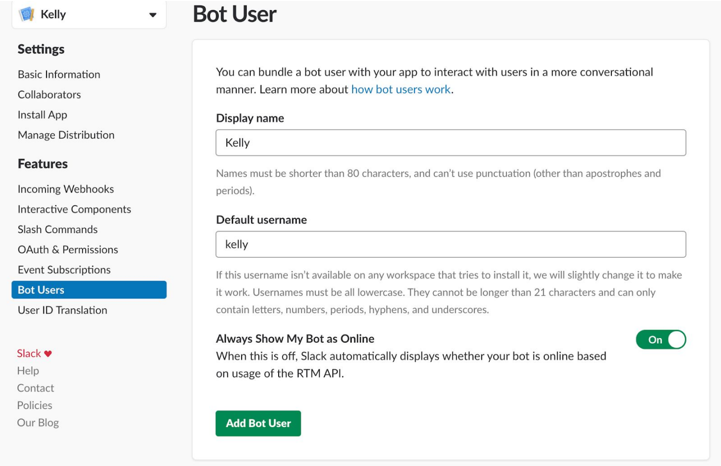
Incoming Webhooks: Post messages from external sources into Slack.

Interactive Components: Add buttons to your app's messages, and create an interactive experience for users.

Slash Commands: Allow users to perform app actions by typing commands in Slack.

Event Subscriptions: Make it easy for your app to respond to activity in Slack.

Step3 - Create a Bot User



You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about how bot users work.

Display name: Kelly

Names must be shorter than 80 characters, and can't use punctuation (other than apostrophes and periods).

Default username: kelly

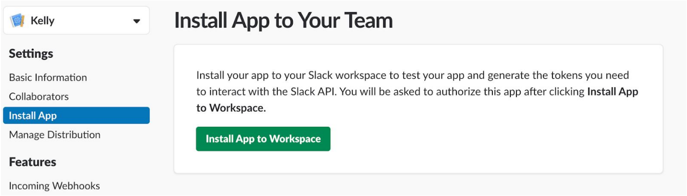
If this username isn't available on any workspace that tries to install it, we will slightly change it to make it work. Usernames must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores.

Always Show My Bot as Online: On

When this is off, Slack automatically displays whether your bot is online based on usage of the RTM API.

Click on the bot user features on the left panel of the app page and fill in the details. Put the **Display Name** and **Default username** as **[your zID]_bot** (i.e. ***z5551234_bot***). Please make sure to turn on the toggle option to show your bot always online. Once the details are provided, click "Add Bot User"

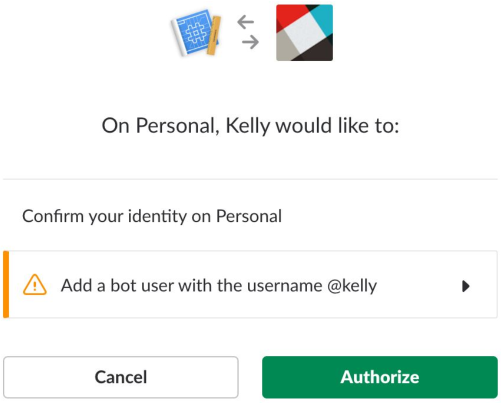
Step4 - Install Bot on your workspace



Install App to Your Team

Install your app to your Slack workspace to test your app and generate the tokens you need to interact with the Slack API. You will be asked to authorize this app after clicking **Install App to Workspace.**

Now, you are ready to install the app on your workspace. Navigate to the Install App panel and click on the button "Install App to Workspace".



On Personal, Kelly would like to:

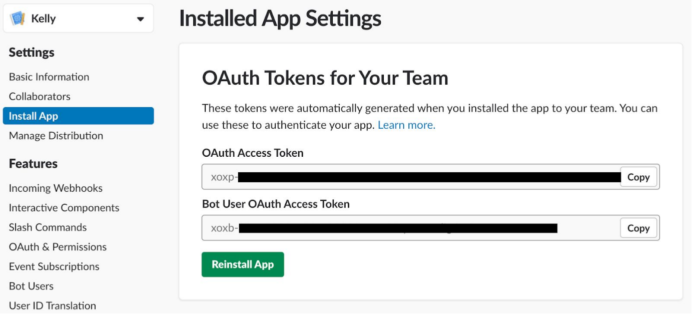
Confirm your identity on Personal

⚠ Add a bot user with the username @kelly

Click "Authorize". That is all. Your slack bot is installed and you can access it from slack app.

Step5 - Make a note of bot tokens

Now you will be redirected to the bot token page as shown below. Your bot need **"Bot User OAuth Access Token"** and **"Verification Token"**



Installed App Settings

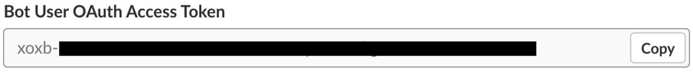
OAuth Tokens for Your Team

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. Learn more.

OAuth Access Token: xoxp-[redacted]

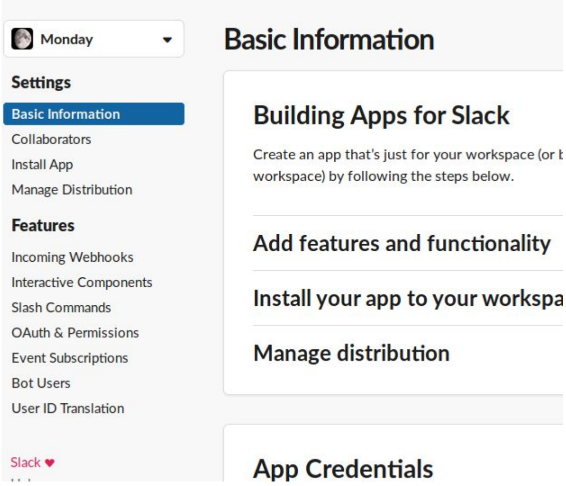
Bot User OAuth Access Token: xoxb-[redacted]

Make a note of the **"Bot User OAuth Access Token"**. This token will be needed to access your slack app from python. This token will be put on **config.py** as **"SLACK_BOT_TOKEN"**



Bot User OAuth Access Token: xoxb-[redacted]

Next, go to **"Basic Information"** tab and look for **"Verification Token"** under **"App Credentials"** section" as shown below.



Monday

Settings

- Basic Information
- Collaborators
- Install App
- Manage Distribution

Features

- Incoming Webhooks
- Interactive Components
- Slash Commands
- OAuth & Permissions
- Event Subscriptions
- Bot Users
- User ID Translation

Slack ♥

Basic Information

Building Apps for Slack

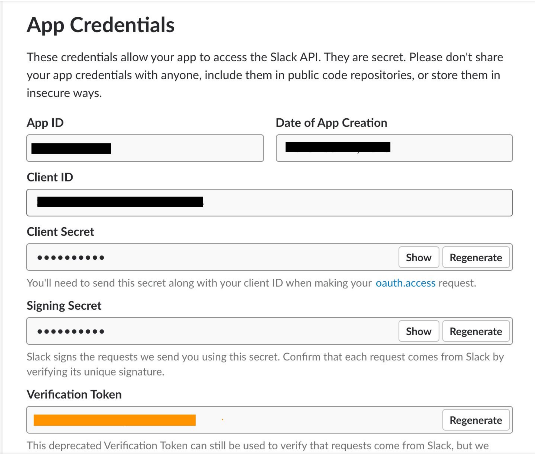
Create an app that's just for your workspace (or t workspace) by following the steps below.

Add features and functionality

Install your app to your workspa

Manage distribution

App Credentials



App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, including them in public code repositories, or store them in insecure ways.

App ID: [redacted]

Date of App Creation: [redacted]

Client ID: [redacted]

Client Secret: •••••••••• Show Regenerate

You'll need to send this secret along with your client ID when making your oauth.access request.

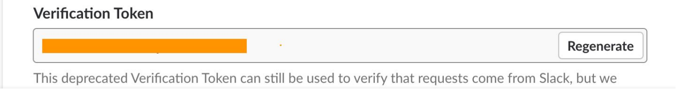
Signing Secret: •••••••••• Show Regenerate

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token: [redacted] Regenerate

This deprecated Verification Token can still be used to verify that requests come from Slack, but we

Make a note of the **"Verification Token"**. This token will be put on **config.py** as **"SLACK_VERIFICATION_TOKEN"**



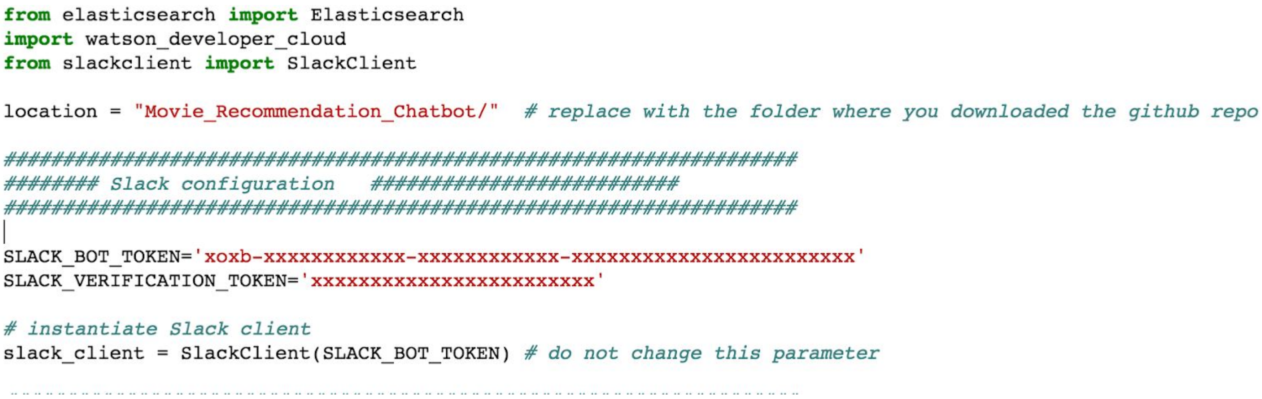
Verification Token

[redacted] Regenerate

This deprecated Verification Token can still be used to verify that requests come from Slack, but we

Step6 - Edit the config.py file

The final step is to add your bot tokens in the config.py file.



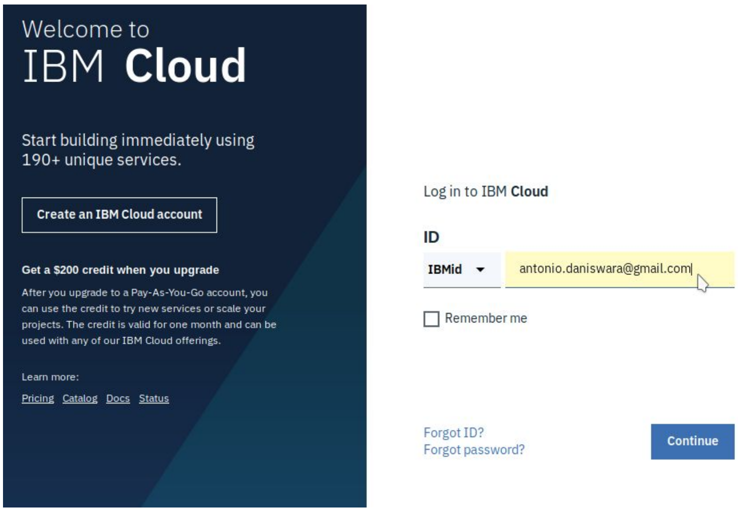
```
from elasticsearch import Elasticsearch
import watson_developer_cloud
from slackclient import SlackClient

location = "Movie_Recommendation_Chatbot/"  # replace with the folder where you downloaded the github repo

###################################################################
######## Slack configuration   ##########################
###################################################################

SLACK_BOT_TOKEN='xoxb-xxxxxxxxxxxx-xxxxxxxxxxxx-xxxxxxxxxxxxxxxxxxxxxxxx'
SLACK_VERIFICATION_TOKEN='xxxxxxxxxxxxxxxxxxxxxxxx'

# instantiate Slack client
slack_client = SlackClient(SLACK_BOT_TOKEN) # do not change this parameter
```

We have completed our slack setup. Have fun!

Act 3 - IBM Watson Conversation setup

Step 1 - Login to IBM Cloud

Go to https://cloud.ibm.com/login, and login with your registed email from week2 tutorial



Welcome to
IBM Cloud

Start building immediately using 190+ unique services.

Create an IBM Cloud account

Get a $200 credit when you upgrade

After you upgrade to a Pay-As-You-Go account, you can use the credit to try new services or scale your projects. The credit is valid for one month and can be used with any of our IBM Cloud offerings.

Learn more:
Pricing Catalog Docs Status

Log in to IBM Cloud

ID

IBMid ▾ antonio.daniswara@gmail.com

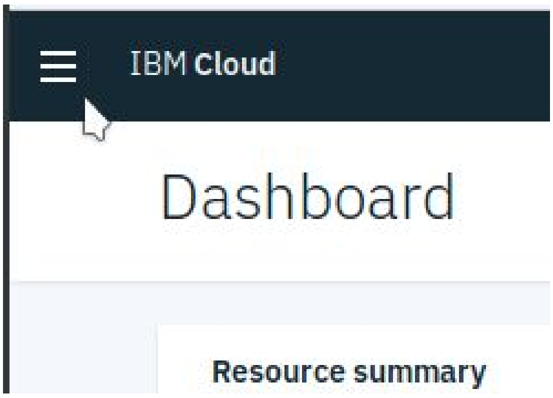
□ Remember me

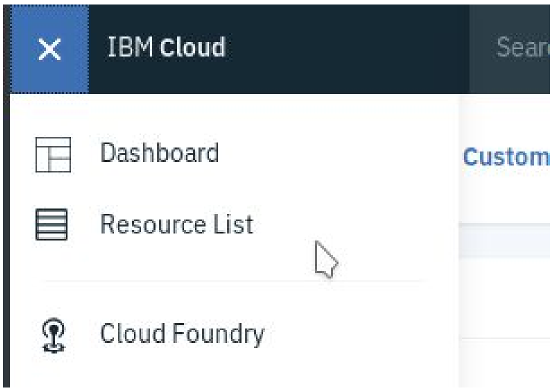
Forgot ID?
Forgot password?

Continue

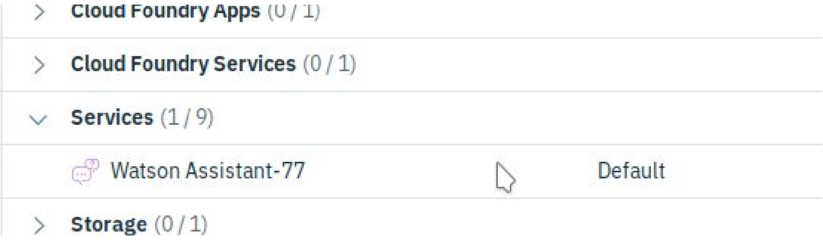
Step 2 - Open the Watson Assistant Resource

Open Navigation Menu Sidebar by clicking menu button (top level corner). Select Resource List.





Open **Services** and select your Watson Assitant resources



Cloud Foundry Apps (0 / 1)

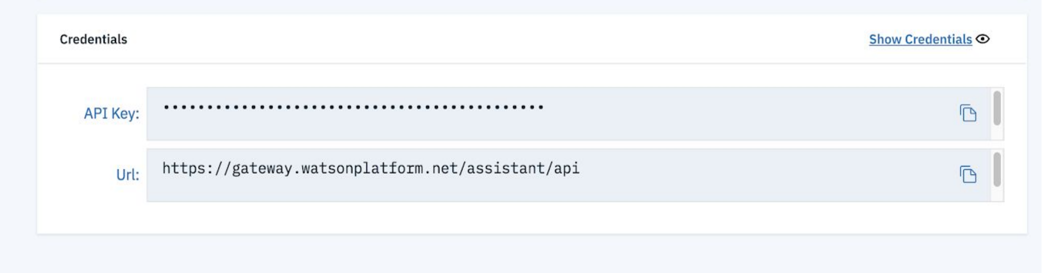
Cloud Foundry Services (0 / 1)

Services (1 / 9)

Watson Assistant-77 Default

Storage (0 / 1)

Copy **URL** under **Credentials** section into **config.py** file **service** object .
Now you can click on **"Launch tool"** and start adding skills for your bot.



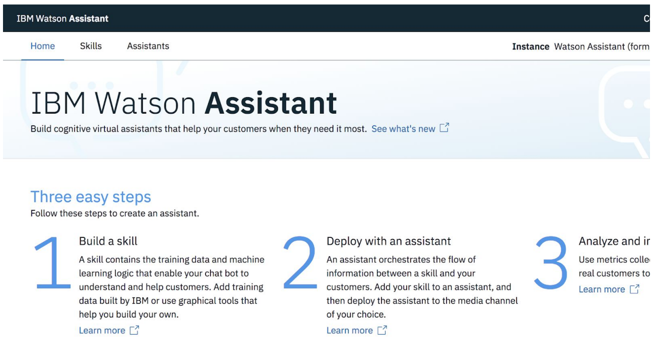
Credentials Show Credentials

API Key: ••

Url: https://gateway.watsonplatform.net/assistant/api

Step3 - Create a Bot workspace and add skills

This is where you start your bot workspace and start adding skills. The steps below will guide you to accomplish that.



IBM Watson **Assistant** C

Home Skills Assistants **Instance** Watson Assistant (form

IBM Watson **Assistant**

Build cognitive virtual assistants that help your customers when they need it most. See what's new

Three easy steps

Follow these steps to create an assistant.

1 Build a skill

A skill contains the training data and machine learning logic that enable your chat bot to understand and help customers. Add training data built by IBM or use graphical tools that help you build your own.

Learn more

2 Deploy with an assistant

An assistant orchestrates the flow of information between a skill and your customers. Add your skill to an assistant, and then deploy the assistant to the media channel of your choice.

Learn more

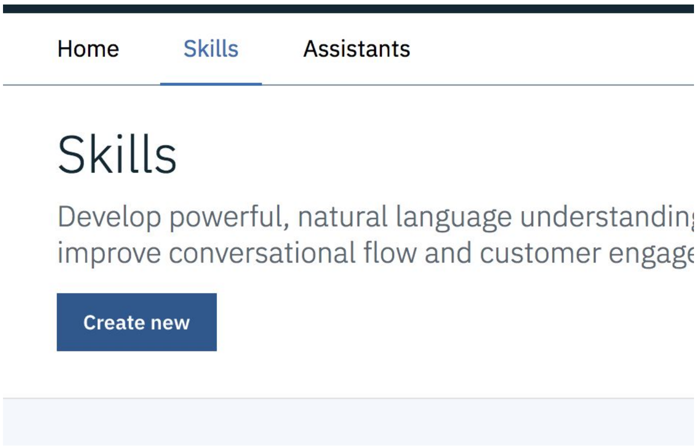
3 Analyze and ir

Use metrics colle

real customers to

Learn more

Click on **Skills** tab, and Create New



Home Skills Assistants

Skills

Develop powerful, natural language understandin

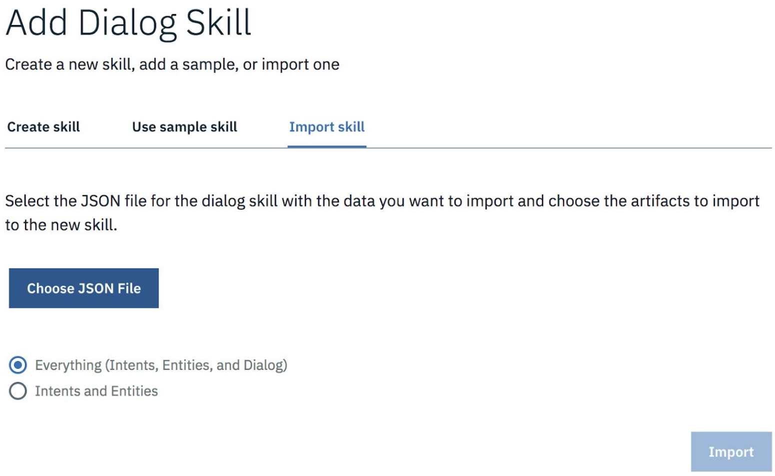
improve conversational flow and customer engage

Create new

On this tutorial we will import premade skill from **bot.json**

Step4 - Import a existing workspace to your Watson conversation.

In the add dialog skill box shown above, there is a tab to import skill. This feature can be used to upload a workspace to your resource. You can use **bot.json** file in this project folder to be imported



Add Dialog Skill

Create a new skill, add a sample, or import one

Create skill Use sample skill Import skill

Select the JSON file for the dialog skill with the data you want to import and choose the artifacts to import to the new skill.

Choose JSON File

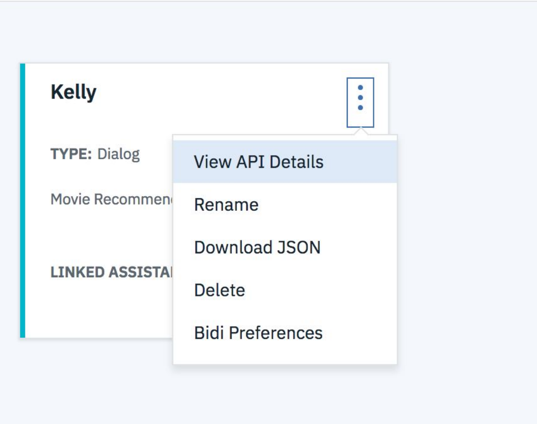
● Everything (Intents, Entities, and Dialog)

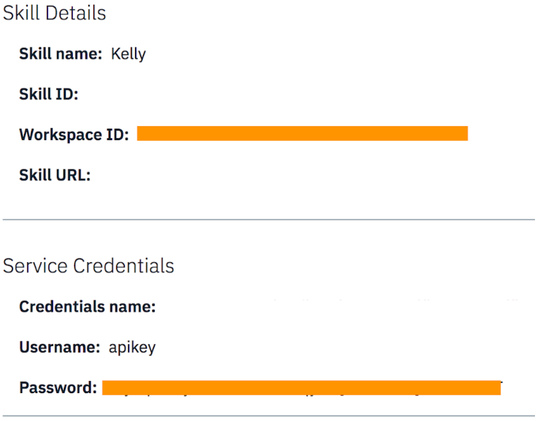
○ Intents and Entities

Import

Step 5 - Accessing the API keys for python config files

Once you are done with adding skills to your bot, it is time to deploy it. We need the keys to add to the configuration python file.





Skill Details

Skill name: Kelly

Skill ID:

Workspace ID:

Skill URL:

Service Credentials

Credentials name:

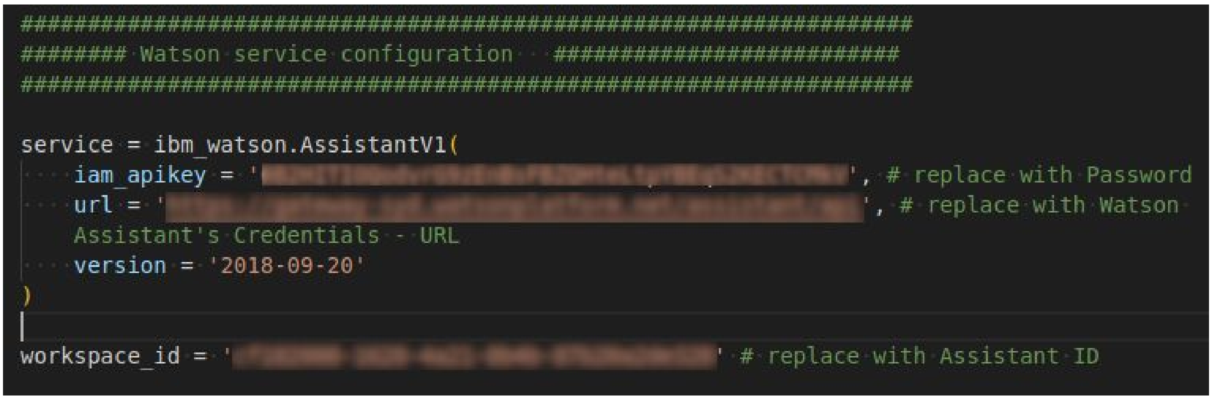
Username: apikey

Password:

Get the API details for your bot, by clicking on the "View API Details" and you need to grab two pieces of information to add to configuration file. This information is highlighted in orange color.

Step 6 - Edit the config.py with watson configuration

Copy the **Workspace ID** (Skill Details) to **workspace_id** in **config.py.**
Copy **Password** (service Credentials) to **iam_apikey** in **config.py**



```
###################################################################
######## Watson service configuration   ##########################
###################################################################

service = ibm_watson.AssistantV1(
    iam_apikey = '[redacted]', # replace with Password
    url = '[redacted]', # replace with Watson
    Assistant's Credentials - URL
    version = '2018-09-20'
)

workspace_id = '[redacted]' # replace with Assistant ID
```

For more information about to build a client application, pleas read the documentation in the links below:

1. https://console.bluemix.net/apidocs/assistant?language=python
2. https://console.bluemix.net/docs/services/assistant/api-client.html#building-a-client-application

Act 4 - Preparing Work Enviroment

Open **Anaconda Prompt**

Go to the project folder

```
cd path\to\folder
```

Create virtual environment

```
conda create -n tutorial4 python=3.6.5
```

Activate virtual environment

```
source activate tutorial4
```

Install packages from requirements.txt

```
pip install -r requirements.txt
```

Act 5 - Preparing Data

Note: you just need do this activity once

From project folder, run **data_prep.py** with this command

```
python data\data_prep.py
```

After the python file stop running, new **metadata_prep.cvs** will created
Note: you can open the file using text editor to see how it works

Run this command to create **onetime.txt**

```
python nlp\nlp_solutions\onetime_run_file.py
```

Act 6 - Running & Testing The Bot

Step 1 - Run your Bot backend

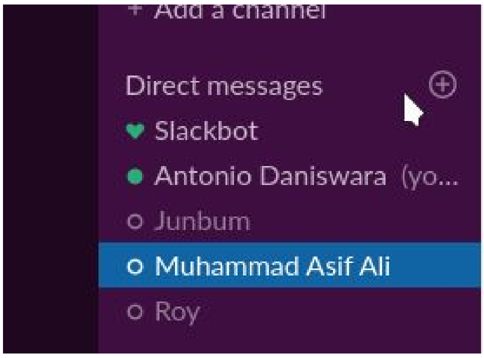
After You make sure all configuration (**config.py**) is correct and 2 files (**metadata_prep.cvs & onetime.txt)** created, you need to run the code below (from project directory):

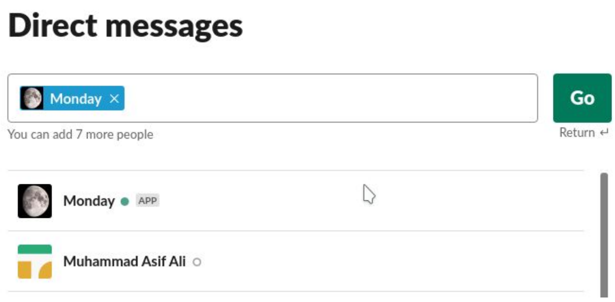
```
python main.py
```

[screenshoot here]

Step 2 - Find your Bot

Don't close the terminal. Login to your Slack Workspace. Click on **Direct Messages** and type your bot name (i.e. **z5551234_bot**).

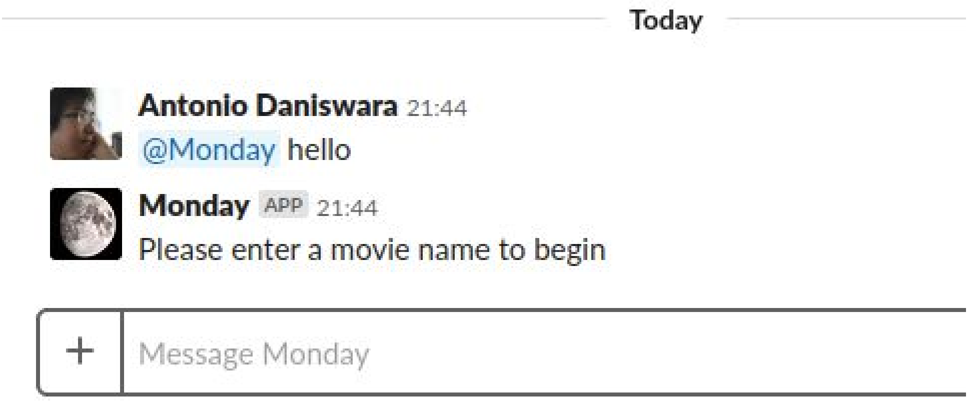




Step 3 - Interact with your Bot

Note : In this example you need to mention your bot (i.e **@z5551234_bot**) to interact.

First try to say hello to your bot

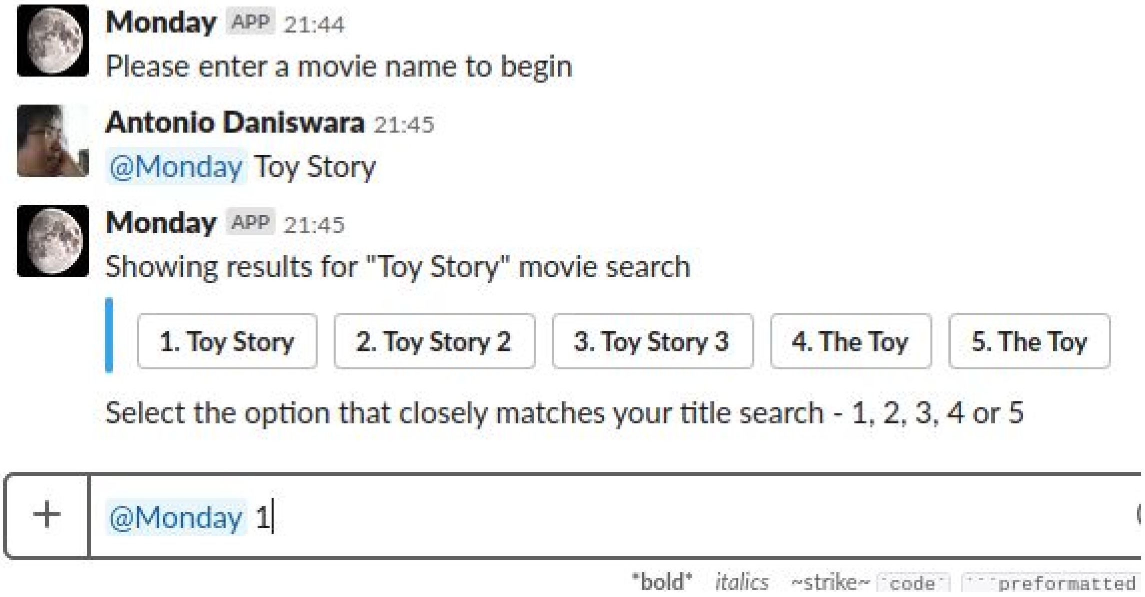


Today

Antonio Daniswara 21:44
@Monday hello

Monday APP 21:44
Please enter a movie name to begin

After your bot asking movie title, you can put any movie you want to know (i.e **Toy Story**)



Monday APP 21:44
Please enter a movie name to begin

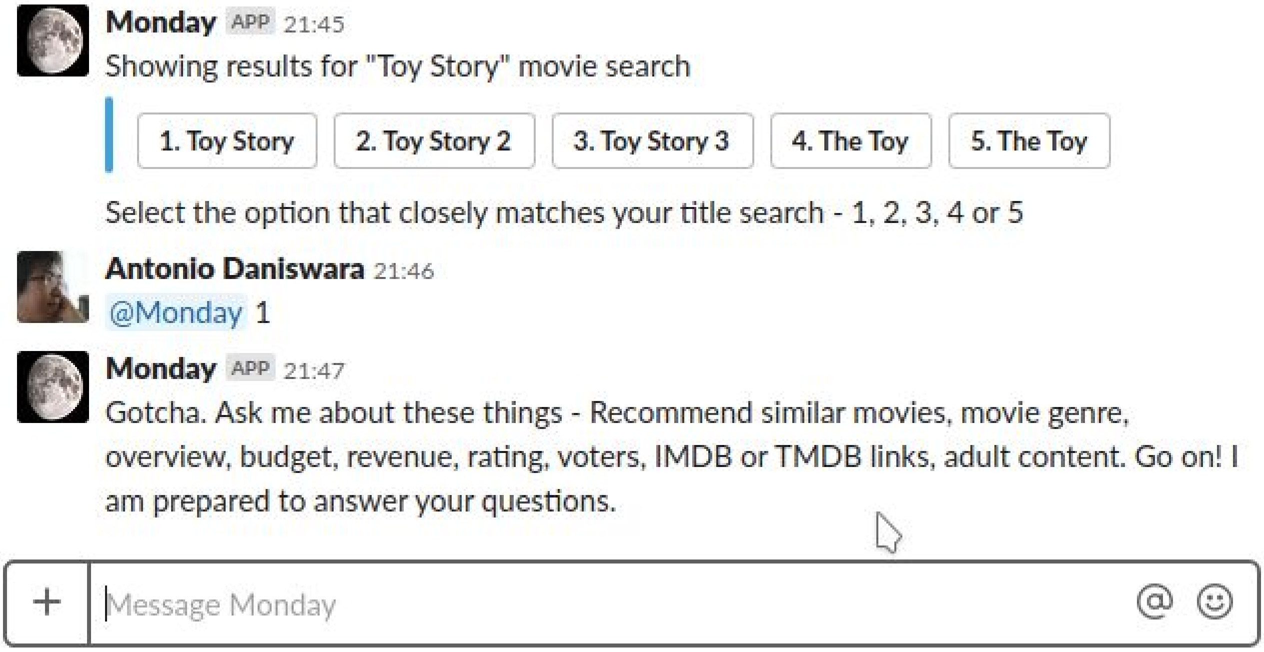
Antonio Daniswara 21:45
@Monday Toy Story

Monday APP 21:45
Showing results for "Toy Story" movie search

Select the option that closely matches your title search - 1, 2, 3, 4 or 5

@Monday 1

Your bot will try to find the title from dataset. It will give 5 similiar results. Type the number



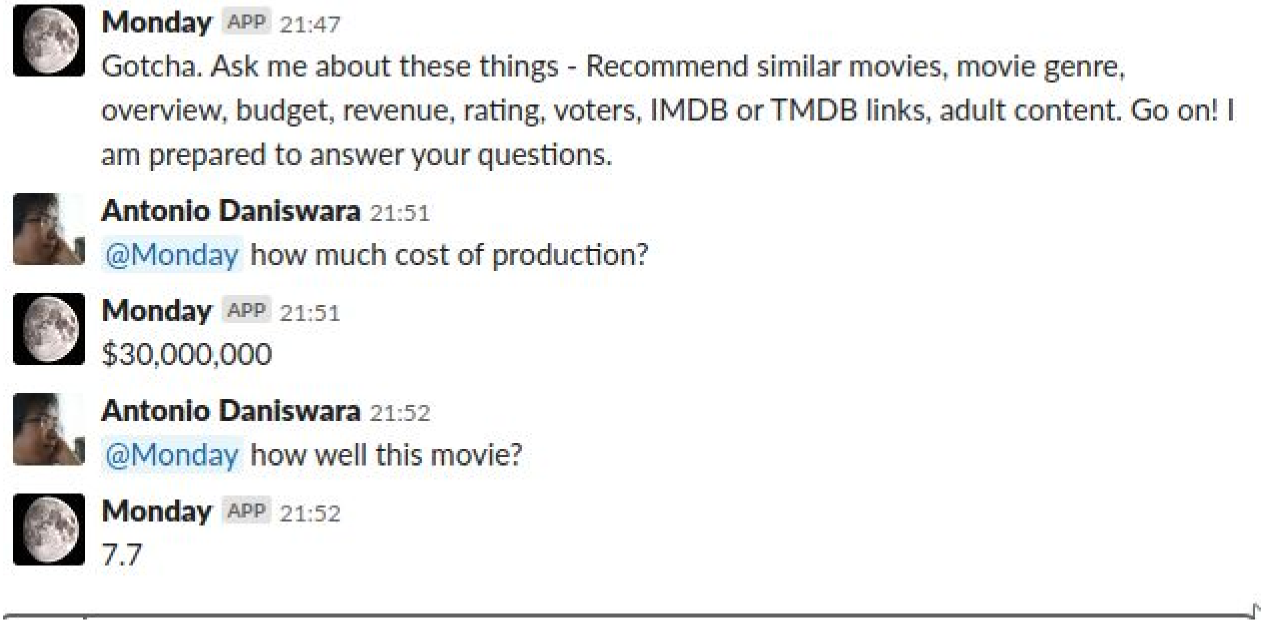
Monday APP 21:45
Showing results for "Toy Story" movie search

Select the option that closely matches your title search - 1, 2, 3, 4 or 5

Antonio Daniswara 21:46
@Monday 1

Monday APP 21:47
Gotcha. Ask me about these things - Recommend similar movies, movie genre, overview, budget, revenue, rating, voters, IMDB or TMDB links, adult content. Go on! I am prepared to answer your questions.

Now, Your bot will save the movie information in memory. You can ask anything about the movie (using Natural Languange).



Monday APP 21:47
Gotcha. Ask me about these things - Recommend similar movies, movie genre, overview, budget, revenue, rating, voters, IMDB or TMDB links, adult content. Go on! I am prepared to answer your questions.

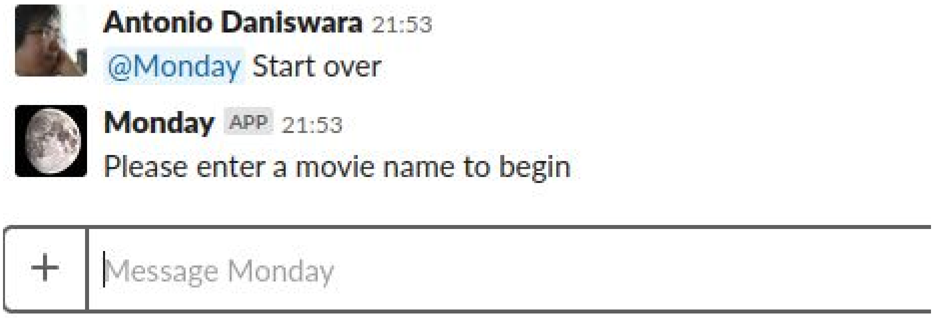
Antonio Daniswara 21:51
@Monday how much cost of production?

Monday APP 21:51
$30,000,000

Antonio Daniswara 21:52
@Monday how well this movie?

Monday APP 21:52
7.7

To look new movie information, you can say “start over” to your bot



Antonio Daniswara 21:53
@Monday Start over

Monday APP 21:53
Please enter a movie name to begin

End Of Tutorial