

SJSU ChatMe Chatbot

A Slack based app for SJSU related queries

Arpit Mathur (012428806)
Software Engineering Department
San Jose State University
San Jose, CA, USA
arpit.mathur@sjsu.edu

Harsha Muktamath
Software Engineering Department
San Jose State University
San Jose, CA, USA
harsha.muktamath@sjsu.edu

This paper describes SJSU ChatMe slackbot which can be used to ask day to day questions in SLACK interface regarding location, courses, library time and weather-related questions specifically about SJSU. This helps in making SJSU students life easier in getting to know the university building location, courses, weather with a click of a button. Students can visit <https://distinct-cross.glitch.me/> to onboard the app to their channel and start asking questions instantly.

I. Introduction

This paper presents a simple slackbot leveraging the power of IBM Watson with services like Conversation and Weather and integrated in glitch server for botkit usage. Botkit is designed to ease the process of designing and running useful, creative bots that live inside messaging platforms. Bots are applications that can send and receive messages, and in many cases, appear alongside their human counterparts as users. Glitch is made by Fog Creek Software, which is a platform and an environment to develop and host application. In the current project, botkit is integrated with glitch server for easy and fast development of slackbot.

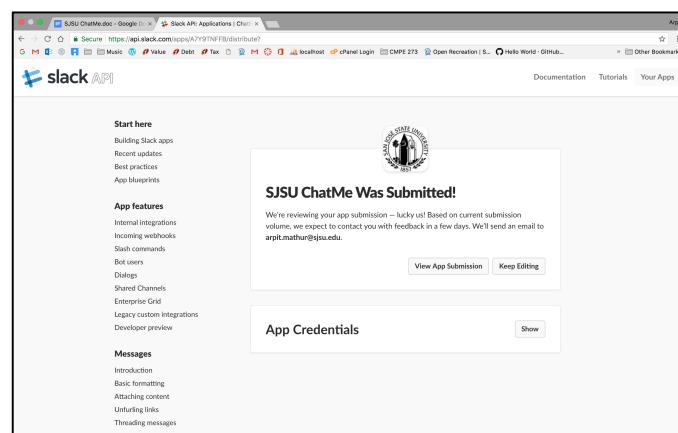


Fig. 1. App Submitted for review to be added in the list of official apps.

Jainul Patel (012463737)
Software Engineering Department
San Jose State University
San Jose, CA, USA
jainul.patel@sjsu.edu

The slackbot is developed to answer various questions like info about the professor, about the classes , about various locations in the campus etc.

The chatbot created has been sent for review to be included in the list of official Slack App Directory (<https://slack.com/apps/A7Y9TNFFB-sjsu-chatme>).

II. Choosing the Messaging Service.

In order to reach a broader audience, the messaging service should have cross platform compatibility. So, a messaging service that is well integrated with all modes of communication i.e. it has a good desktop application, web application and has mobile app for all platforms like Android and iOS needed to be used. Messaging services like WhatsApp, Slack, Facebook Messenger and Google Hangouts were considered.

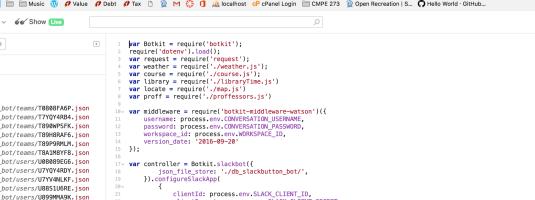
Next, we wanted the service to provide good API support for our chatbot to work. The Slack's Real Time Messaging API service in which we can subscribe to various events like new message in a group or new message in a private message looked promising. These events could be subscribed via Webhooks. These Webhooks send data to Websocket URLs where our backend program could be plugged in for further processing.

Thus, Slack was chosen as the messaging service for our chatbot.

III. Hosting the bot.

Slack's Real Time Messaging (RTM) API sends and receives messages through Webhooks which are provided through Websocket URLs. To host this Websocket URL Glitch (made by Fog Creek Software) is used. This provides a

24/7 running server for the chatbot to run and listen to any message and reply to it accordingly.



The screenshot shows a browser developer tools console with the URL <https://github.com/edw/district-cross-path/watson-slack.js>. The code is a single large file containing various functions and configurations for integrating Watson with Slack. It includes imports for Botkit, Botkit Slack Adapter, and Watson modules like Watson Slack, Watson Weather, and Watson Text-to-Speech. The code sets up a Slack bot, configures it with an access token, and defines several functions for handling messages, tracking users, and invoking Watson services. A 'New File' button is visible at the top left, and a 'New Tab' button is at the bottom right.

Fig. 2. Backend of bot hosted on Glitch servers

IV. How to build

Clone the directory from

<https://github.com/SJSU272LabF17/Project-Team-28>

Three different sets of credentials are required for building this chat bot. Add all these credentials into .env file.

- 1) Watson Conversation
 - 2) IBM Weather Company Data
 - 3) Slack App Credentials

Steps to get Watson Service Credentials:

- 1) Login to Bluemix at <https://bluemix.net>
 - 2) Create an instance of the service
 - 3) In the Bluemix catalog, select the Watson Conversation Service.

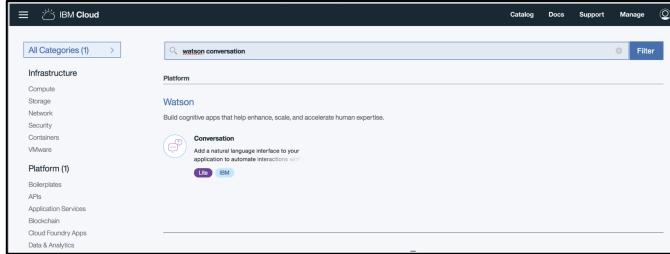


Fig. 3. IBM Watson Service.

- 4) Type a unique name for the service instance in the service name field and leave the other default values as it is.
 - 5) Now Click “Create” and on the Dashboard, your service instance is listed.

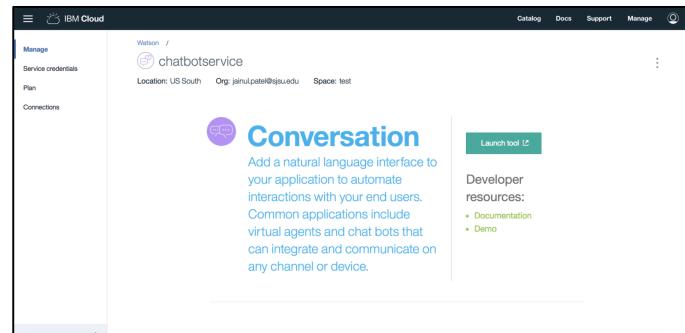


Fig. 4. Conversation Dashboard.

- 6) From the service dashboard navigation pane, click on Service Credentials.

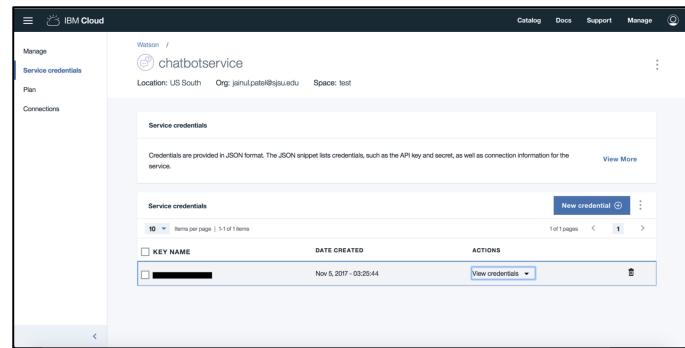


Fig. 5. Conversation Credentials

- 7) Under Actions, click “View Credentials”.
 - 8) Copy username and password and save it to .env file.

Steps to get weather company data credentials:

- 1) Login to Bluemix at <https://bluemix.net>
 - 2) Create an instance of the service
 - 3) In the Bluemix catalog, select the Weather Company Data.

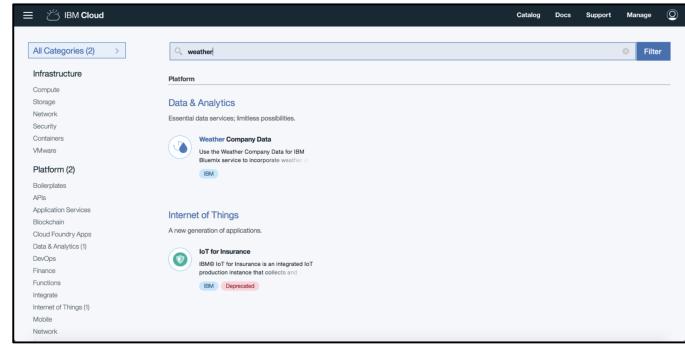


Fig. 6. IBM Weather Service.

- 4) Type a unique name for the service instance in the service name field and leave the other default values as it is.
- 5) Now Click “Create” and on the Dashboard, your service instance is listed.
- 6) From the service dashboard navigation pane, click on Service Credentials.
- 7) Under Actions, click “View Credentials”.
- 8) Copy username and password and save it to .env file.

Steps to get Slack credentials:

- 1) Signup with slack
- 2) Create a workspace
- 3) From <https://api.slack.com/apps> create a new app

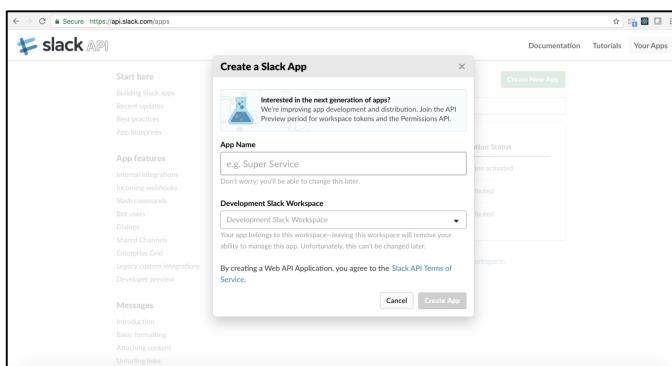


Fig. 7. Creating Slack App.

- 4) From Basic information tab, copy App credentials and paste it to .env file

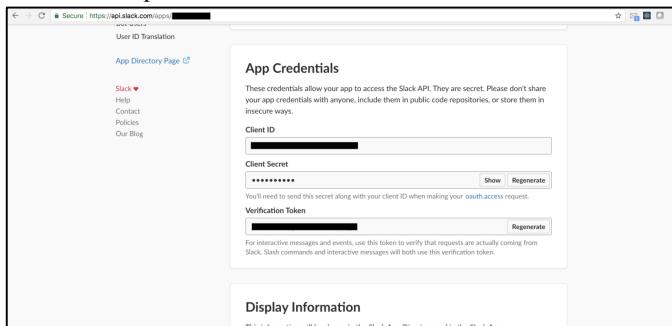


Fig. 8. Slack Credentials.

Once all three set of credentials are generated, turn on the BotKit app by passing the values of clientId, clientSecret and PORT=3000 node via the command line provided the Botkit Slack Starter Kit is being used.

Slack and Botkit both use the oauth authentication system which allow bots access of connecting, reading and sending messages to Slack teams.

From the Slack app settings, click on the "Oauth & Permissions" tab and under Redirect URLs, add: https://my-bot-url/oauth, then click save.

Click on the "Bot Users" tab and assign a name for the bot which will be used by default when the new bot on user's team is created. Enable the "Always Show My Bot Online" option.

Under the Request URL, click on the "Interactive Components" tab and add https://my-bot-url/slack/receive, then click save.

Enable event subscriptions to start receiving messages in the chatbot. In order to facilitate this, add <https://my-bot-url/slack/receive> under Request URL. Once the URL is added, the endpoint configuration verification is done by Slack. For this verification, the Botkit application must be in a running state and the URL must be accessible.

After the verification, click "Add Bot User Event", and using the dropdown that appears, select all of the message events: message.groups, message.channels, message.mpim, message.im which informs Slack to send all messages to the bot that are sent in any channel or group in which the bot has been added.

Switch on the "Enable Events" and the bot is now ready to answer your questions.

Now once the application is up and running, you can login and add the bot and the bot is now ready to answer your queries.

V. User Onboarding

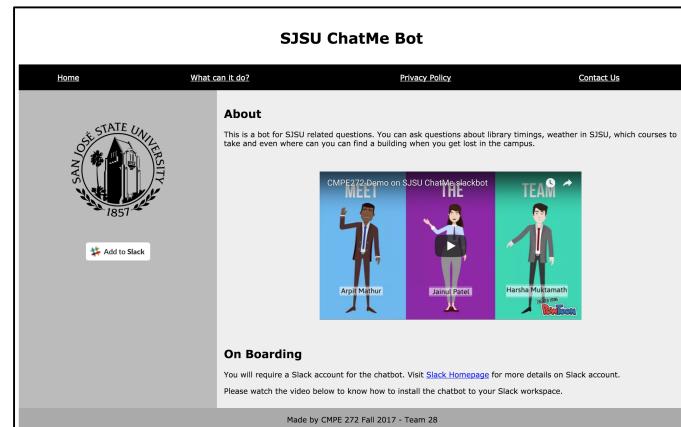


Fig. 9. User Onboarding “Add to Slack button”

User Onboarding is one of the critical aspects of our project and we have made this process very user friendly, simple and quick. The prerequisites required to get going in using our ChatMe slackbot is just the Slack app on his/her mobile phone or laptop and user is all set to start slacking with our ChatMe bot. We have developed a simple web page where user can checkout the onboarding video and also get on board in using ChatMe with just a click of button. (<https://www.youtube.com/watch?v=nr5Vdsu8WcU>)

More details are provided at <https://distinct-cross.glitch.me/> and just click on the “Add to Slack” button to integrate our bot in your pre installed Slack app.

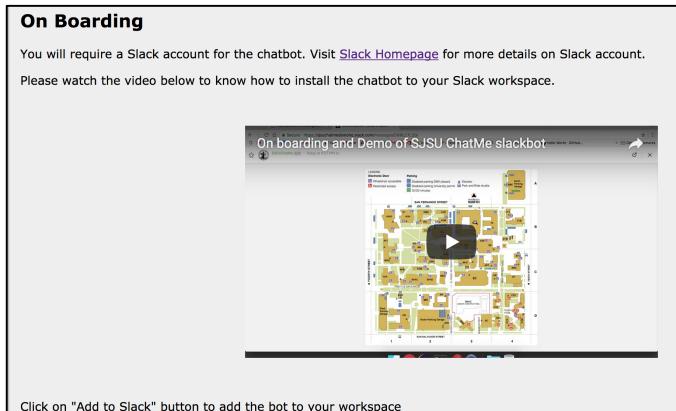


Fig.10 User Onboarding “ Video”

VI. Use Cases

VII.

There are many things you can ask the bot:

- 1) Confusion about courses to take?
 - a) Which are the subjects related to Data science
 - b) Subjects in cloud computing



Fig. 11. Bots reply for Location Queries.

- 2) Can not find a building in campus. Just ask the bot.
 - a) Where is chase ATM?
 - b) Locate engineering building
 - c) How to go to tower hall

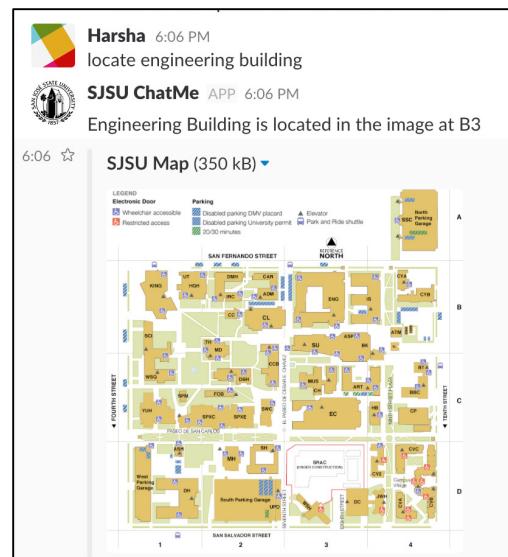


Fig. 12. Bots reply for Location Queries.

- 3) Want to know about a professor?
- a) Tell me about Prof Rakesh Ranjan
- b) Office hours of Prof Dan Harkey?

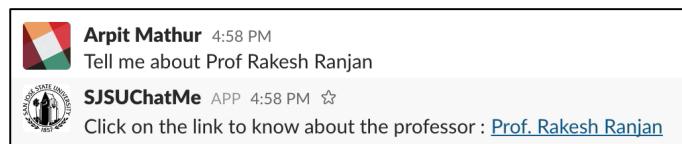


Fig. 13. Professor related queries.

- 4) Want to know real time weather in campus
- a) How is todays weather

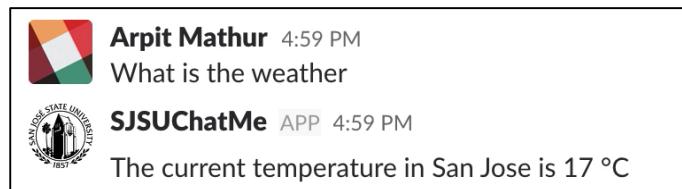


Fig. 14. Weather Queries.

- 5) Want to find out if you can go to the library?
- a) Is library open now?
- b) When can I return a book to the library?

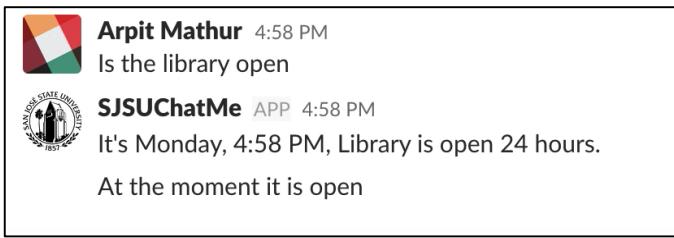


Fig. 15. Library Queries.

Acknowledgment

The authors would like to thank Prof. Rakesh Ranjan, Department of Computer Engineering, San Jose State University, California for guiding us in this project and giving us valuable inputs during the course of the project. A special mention and appreciation to the professor for giving us a Free trial version of Bluemix account for 6 months which helped in us choosing some of the services like Conversation and Weather company data on IBM Watson.

References

- [1] <https://github.com/IBM-Bluemix/phonebot> , ‘Slackbot using IBM Watson and Twilio to make phone calls via slack commands’ , 2017 [Accessed: 23- Nov- 2017].
- [2] www.npmjs.com/package/botkit-middleware-watson , ‘Use IBM Watson’s Conversation service to chat with your Botkit-powered Bot! ’ , 2017 [Accessed: 25- Nov- 2017].
- [3] <https://github.com/ibm-watson-data-lab/slack-analytics> , ‘Searching Slack with IBM Graph’ , 2017 [Accessed: 26- Nov- 2017].