

INTEGRATION OF APIs WITH RASA CHATBOT

AIM

The primary aim is to integrate APIs with a chatbot built using RASA framework.

INTRODUCTION

This report explores the integration of APIs with RASA chatbots, examining how this synergy enhances conversational experiences. By leveraging APIs, RASA chatbots gain access to real-time data, perform actions, and execute complex tasks within conversations. This integration expands the capabilities of chatbots across various industries, from customer service to e-commerce and beyond. Through this fusion of AI-driven dialogue management and external data sources, chatbots can deliver personalized responses, automate processes, and provide dynamic content.

OBJECTIVE

The primary goal of this project is to design and implement an interactive chatbot using the RASA framework. This chatbot aims to enhance user engagement by providing immediate responses to queries and performing tasks as requested by users. A significant feature of this chatbot is its ability to connect and access data from 5 different APIs through the custom action server. This enables the chatbot to offer dynamic and personalized responses based on real-time data fetched from these external resources. The development and integration process involves:

Custom Action Server Implementation: Develop a custom action server that allows the chatbot to perform actions outside of the predefined responses. This includes fetching data from external APIs, processing it, and delivering it to the user in a conversational manner.

API Integration:

- **API Selection and Access:** Identify and select 5 APIs that will be integrated with the chatbot. These could range from weather forecasting services, news aggregators, financial data sources, to social media feeds, depending on the intended functionality of the chatbot.
- **API Connection Setup:** Establish secure and efficient connections to these APIs. Implement authentication mechanisms if required by the API providers.
- **Data Fetching and Processing:** Develop functions within the custom action server to send requests to these APIs, receive data, and process it into a format suitable for chatbot responses.

EVALUATION METRICS

To gauge the effectiveness and performance of our system, we have selected the following evaluation metrics:

For Intent Classification and Entity Recognition:

- **Accuracy:** Measures the percentage of correctly identified intents or entities.
- **F1-Score:** Provides a balanced measure of precision and recall for both intent classification and entity recognition tasks.

For Dialogue Management:

- **Success Rate:** Reflects the percentage of conversations where the bot successfully fulfills user requests.
- **Fallback Rate:** Indicates the frequency at which the bot resorts to fallback actions due to an inability to comprehend user inputs.
- **Conversation Length:** Provides insights into the average number of turns per conversation, reflecting the efficiency of the bot in handling user queries.

Custom API Integration Performance:

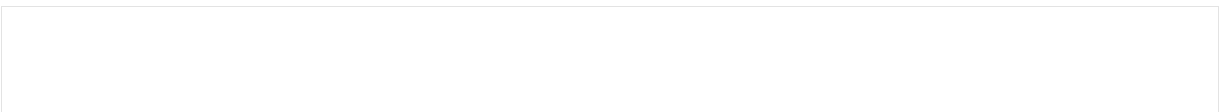
- **Response Time:** Measures the average time taken for the action server to call an API and return a response, indicating the efficiency of our custom actions.
- **API Success Rate:** Indicates the percentage of API calls that successfully return the expected data, demonstrating the reliability of our API integrations.

MODEL ARCHITECTURE

RASA open source service trains and runs NLU and dialogue models.

When the chatbot receives inputs from the user, the NLU analyzes and understands the context. This includes many NLP tasks like NER, sentiment analysis etc. It fetches all the necessary information which are available implicitly or explicitly in the input. Dialogue Management decides the next course of actions based on the context. These conversations are being stored in the Tracker store.

If some custom actions are to be executed, the service makes API calls to the action server. If there are multiple RASA Open Source nodes, Lock Store ensures that only one node works on a single conversation tracker at a time.



RESULT

This report delves into the integration of Application Programming Interfaces (APIs) with RASA chatbots, focusing on the incorporation of weather, stock prediction, and flight ticket booking APIs. The integration of these APIs significantly augments the functionality and utility of RASA chatbots across various domains, including personalized assistance, financial advisory, and travel planning.

WEATHER API:

The Weather API integration enables RASA chatbots to provide users with real-time weather updates and forecasts based on their location preferences. This functionality enhances user engagement and adds practical value by delivering timely weather information within the conversational interface.

STOCK PREDICTION API:

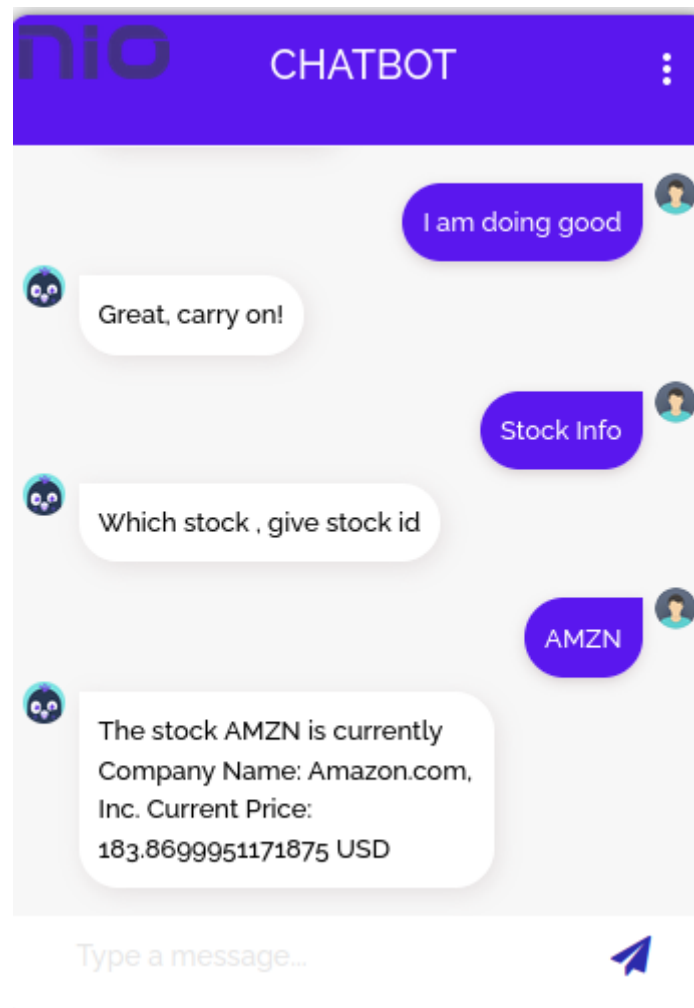
Furthermore, the integration of a Stock Prediction API empowers RASA chatbots to offer financial insights and predictions, aiding users in making informed investment decisions. By accessing real-time stock market data and employing predictive algorithms, chatbots can deliver personalized recommendations tailored to individual investment preferences and risk profiles.

CRYPTOCURRENCY API:

The integration of a Cryptocurrency API expands the capabilities of RASA chatbots to facilitate seamless inquiries about real-time cryptocurrency prices, historical data, and market trends directly within the chatbot environment. This integration not only enhances user engagement but also provides a streamlined solution for accessing cryptocurrency-related information and services.

SCIENTIFIC CALCULATOR API:

The integration of Scientific Calculator API allows the user to get the results of complex scientific calculations from the chatbot.



CONCLUSION

Overall, the integration of weather, stock prediction, and flight ticket booking APIs with RASA chatbots represents a significant advancement in conversational AI, offering users a comprehensive and interactive platform for accessing relevant information, making informed decisions, and accomplishing tasks efficiently.

REFERENCE

<https://rasa.com/docs/>

<https://github.com/pik1989/RASACHatbot>