**WIKIBot**

**Welcome Handler :**

response = Map();

response.put("text","Welcome to the Assistant Bot! Hi there! 😊 I'm here to assist you with your queries. Explore, search, and learn with ease. What would you like to do today?");

card = Map();

card.put("title","Welcome to the Assistant Bot!");

card.put("thumbnail","https://www.icegif.com/wp-content/uploads/2023/10/icegif-17.gif");

card.put("theme","modern-inline");

buttons = list();

button1 = Map();

button1.put("label","Search / Wiki");

button1.put("action","search\_wiki");

buttons.add(button1);

button2 = Map();

button2.put("label","Help");

button2.put("action","help");

buttons.add(button2);

card.put("buttons",buttons);

response.put("card",card);

return response;

**Mention Handler :**

response = Map();

if(message.contains("Wikipedia Bot") || message.contains("bot"))

{

text = "Hello! I’m your friendly Wikipedia Bot. Type `/wiki(topic)` to search for any topic!";

}

else if(message.contains("help"))

{

text = "You can use `/wiki(topic)` to search Wikipedia. Try `/wiki(papaya)` to get started.";

}

else

{

// Handle other mentions or unrecognized input

text = "How can I assist you today? Type `/help` for more info.";

}

response.put("text",text);

return response;

**Message Handler:**

responseMessage = Map();

botDetails = Map();

botDetails.put("name","wiki");

responseMessage.put("bot",botDetails);

outputText = "";

responseOne = Map();

if(message.containsIgnoreCase("HI") || message.containsIgnoreCase("HEY") || message.containsIgnoreCase("HELLO") || message.containsIgnoreCase("HI THERE") || message.containsIgnoreCase("HEY THERE") || message.containsIgnoreCase("GREETINGS") || message.containsIgnoreCase("WELCOME") || message.containsIgnoreCase("START") || message.containsIgnoreCase("GO") || message.containsIgnoreCase("CAN"))

{

responseOne.put("text","Hi! How can I help you? :grinning:");

suggestions = Map();

suggestionList = List();

suggestion1 = Map();

suggestion1.put("text","/Wiki");

suggestionList.add(suggestion1);

suggestion2 = Map();

suggestion2.put("text","Help");

suggestionList.add(suggestion2);

suggestions.put("list",suggestionList);

responseOne.put("suggestions",suggestions);

return responseOne;

}

else if(message.containsIgnoreCase("help"))

{

text = "Here's how I can assist you:\n\n1️⃣ Use /wiki <word> to search for information.\n2️⃣ Ask me anything for fun and interactive chats.\n3️⃣ Explore random fun topics like sports, games, and more!";

helpResponse = Map();

helpResponse.put("text",text);

return helpResponse;

}

else

{

sanitizedSearchTerm = message.replaceAll("[/\\-\\(\\)\_@#\\^!\\$%\\&\\\*:;`~{}\\[\\]<>?|',.\\\\\"]","").trim().replaceAll("\\s+"," ");

if(sanitizedSearchTerm == "")

{

responseMessage.put("text","Please type a search word after /wiki \nUsage: /wiki <word> \ne.g., /wiki Zoho");

return responseMessage;

}

apiEndpoint = "https://en.wikipedia.org/api/rest\_v1/page/summary/" + sanitizedSearchTerm;

try

{

apiResponse = getUrl(apiEndpoint,false);

}

catch (exception)

{

responseMessage.put("text","Sorry! Unable to fetch data from Wikipedia at the moment. Please try again later.");

return responseMessage;

}

apiResponseMap = apiResponse.toMap();

responseCode = apiResponseMap.get("responseCode");

if(responseCode == 200)

{

wikiResponseData = apiResponseMap.get("responseText");

wikipediaPageUrl = wikiResponseData.get("content\_urls").get("desktop").get("page");

if(wikiResponseData.get("type").equalsIgnoreCase("disambiguation"))

{

responseMessage.put("text","Multiple results found for - " + sanitizedSearchTerm + "\nPlease refine the search term.\nRefer: [Wikipedia.](" + wikipediaPageUrl + ")");

return responseMessage;

}

pageTitle = wikiResponseData.get("title");

pageSummary = wikiResponseData.get("extract");

thumbnail = wikiResponseData.get("thumbnail");

if(thumbnail != null)

{

imageUrl = thumbnail.get("source");

outputText = "### " + pageTitle + "\n\n" + pageSummary + "\n\nSource: [Wikipedia.](" + wikipediaPageUrl + ")";

}

else

{

outputText = "### " + pageTitle + "\n\n" + pageSummary + "\nSource: [Wikipedia.](" + wikipediaPageUrl + ")";

}

}

else if(responseCode == 404)

{

outputText = "Wiki page not found for - " + sanitizedSearchTerm;

}

else

{

outputText = "Sorry! Something went wrong. Please try again later.";

}

}

responseSnippet = Map();

responseSnippet.put("text",outputText);

bot = Map();

bot.put("name","WikiBot");

responseSnippet.put("bot",bot);

card = Map();

responseSnippet.put("card",card);

slidesList = list();

slidesList0 = Map();

slidesList0.put("type","images");

dataList = list();

dataList.add(imageUrl);

slidesList0.put("data",dataList);

slidesList.add(slidesList0);

responseSnippet.put("slides",slidesList);

return responseSnippet;

**Participation Handler :**

response = Map();

if(operation.equals("removed\_from\_thread") || operation.equals("thread\_closed"))

{

return response;

}

botMessage = null;

if(operation.equals("added"))

{

botMessage = "Thanks for adding me $USER\_MENTION. Hi everyone! :bye-bye:\nI will actively participate in this channel from now on.";

}

else if(operation.equals("removed"))

{

botMessage = "I've been removed from this channel. Adiós, everyone! :bye-bye:";

}

else if(operation.equals("message\_sent"))

{

botMessage = "$USER\_MENTION has sent a message.";

}

else if(operation.equals("message\_edited"))

{

botMessage = "$USER\_MENTION has edited a message.";

}

else if(operation.equals("message\_deleted"))

{

botMessage = "Hey there! $USER\_MENTION has deleted a message.";

}

else if(operation.equals("auto\_followed\_thread"))

{

botMessage = "I've started following this thread automatically. Let's keep ourselves informed together!";

}

else if(operation.equals("added\_in\_thread"))

{

botMessage = "Thanks for adding me $USER\_MENTION. Hi everyone! :bye-bye:\nI will actively participate in this thread from now on.";

}

else if(operation.equals("thread\_reopened"))

{

botMessage = "This thread has been reopened. I will actively participate in this thread from now on.";

}

// Using regular expressions to replace the "$USER\_MENTION" in the bot message with the syntax for mentioning a user.

botMessage = botMessage.replaceAll('\' + "$USER\_MENTION","{@" + user.get("id") + "}");

response.put("text",botMessage);

return response;

**command:**

**Execution Handler :**

responseMessage = Map();

botDetails = Map();

botDetails.put("name","wiki");

responseMessage.put("bot",botDetails);

outputText = "";

try

{

sanitizedSearchTerm = arguments.replaceAll("[/\\-\\(\\)\_@#\\^!\\$%\\&\\\*:;`~{}\\[\\]<>?|',.\\\\\"]","").trim().replaceAll("\\s+"," ");

if(sanitizedSearchTerm == "")

{

responseMessage.put("text","Please type a search word after /wiki \nUsage: /wiki <word> \ne.g., /wiki Zoho");

return responseMessage;

}

apiEndpoint = "https://en.wikipedia.org/api/rest\_v1/page/summary/" + sanitizedSearchTerm;

try

{

apiResponse = getUrl(apiEndpoint,false);

}

catch (exception)

{

responseMessage.put("text","Sorry! Unable to fetch data from Wikipedia at the moment. Please try again later.");

return responseMessage;

}

apiResponseMap = apiResponse.toMap();

responseCode = apiResponseMap.get("responseCode");

if(responseCode == 200)

{

wikiResponseData = apiResponseMap.get("responseText");

wikipediaPageUrl = wikiResponseData.get("content\_urls").get("desktop").get("page");

if(wikiResponseData.get("type").equalsIgnoreCase("disambiguation"))

{

responseMessage.put("text","Multiple results found for - " + sanitizedSearchTerm + "\nPlease refine the search term.\nRefer: [Wikipedia.](" + wikipediaPageUrl + ")");

return responseMessage;

}

pageTitle = wikiResponseData.get("title");

pageSummary = wikiResponseData.get("extract");

thumbnail = wikiResponseData.get("thumbnail");

if(thumbnail != null)

{

imageUrl = thumbnail.get("source");

outputText = "### " + pageTitle + "\n\n" + pageSummary + "\n\nSource: [Wikipedia.](" + wikipediaPageUrl + ")";

}

else

{

outputText = "### " + pageTitle + "\n\n" + pageSummary + "\nSource: [Wikipedia.](" + wikipediaPageUrl + ")";

}

}

else if(responseCode == 404)

{

outputText = "Wiki page not found for - " + sanitizedSearchTerm;

}

else

{

outputText = "Sorry! Something went wrong. Please try again later.";

}

}

catch (exception)

{

outputText = "An error occurred. Please try again.";

}

responseSnippet = Map();

responseSnippet.put("text",outputText);

bot = Map();

bot.put("name","WikiBot");

responseSnippet.put("bot",bot);

card = Map();

responseSnippet.put("card",card);

slidesList = list();

slidesList0 = Map();

slidesList0.put("type","images");

dataList = list();

dataList.add(imageUrl);

slidesList0.put("data",dataList);

slidesList.add(slidesList0);

responseSnippet.put("slides",slidesList);

return responseSnippet;

**Connections :**

**Connection name: Wikipedia**

**Deluge Code:**

response = invokeUrl [

url: <url>

type: GET/POST/PUT/DELETE

parameters: <paramMap/string>

connection: "wikipedia"

];

**JSON Code:**

{

"connectionLinkName": "wikipedia",

"connectionName": "Wikipedia",

"serviceName": "wikipedia",

"userAccess": true,

"isUserDefinedService": true,

"sharedBy": "871151366"

}