



# Tworzenie chatbota zintegrowanego z Discord w oparciu o platformę Dialogflow

Karol Kusper, Szymon Jagła, Jakub Kubica



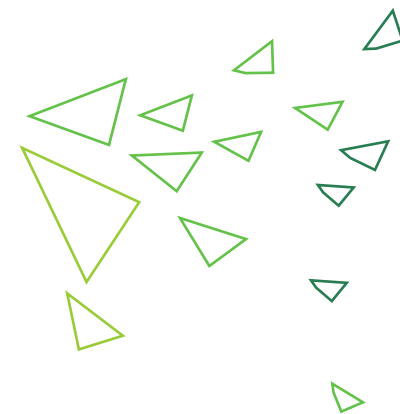
## Spis treści

1. Wstęp
2. Cel platformy i typowe zastosowania
3. Kluczowe funkcje Dialogflow
4. Proces implementacji chatbota
5. Najważniejsze funkcje chatbota
6. Etapy integracji z Discord
7. Integracja z Discord – kod
8. Przykłady interakcji z chatbotem
9. Podziękowania

A meme featuring Superman in his classic blue suit with a red cape, flying forward with a determined expression. Overlaid on his chest is the text "GOOGLE DIALOGFLOW" in a bold, stylized font. "GOOGLE" is in white with a black outline, and "DIALOGFLOW" is in yellow with a black outline.

**GOOGLE  
DIALOGFLOW**

Dialogflow



Dialogflow to zaawansowane narzędzie stworzone przez Google, które umożliwia budowanie aplikacji konwersacyjnych.



## Cel projektu

Stworzenie systemu, który ułatwi dostęp do informacji poprzez chatbota, bez potrzeby odwiedzania strony uczelni lub kontaktu z dziekanatem

Funkcjonalności chatbota:

- Godziny otwarcia dziekanatu: Bot poinformuje o godzinach pracy dziekanatu.
- Informacje o prowadzącym przedmiot: Bot odpowie na pytanie kto uczy danego przedmiotu.
- Propozycje aktywności po zajęciach: Bot może zaproponować co ciekawego można zrobić po zajęciach.
- FAQ: Odpowiedzi na najczęściej zadawane pytania.

# Kluczowe funkcje Dialogflow

## Intencje

Przykłady:

- „Jakie są godziny otwarcia dziekanatu?”
- „Jakie przedmioty są dostępne na kierunku Informatyka?”
- „Kiedy mam zajęcia z przedmiotu Matematyka?”

## Encje

- Encje systemowe:
  - Automatycznie rozpoznają daty, liczby czy godziny (np. „dzień”, „godzina”).
- Encje własne:
  - Nazwy kierunków studiów
  - Nazwy przedmiotów
  - Terminy

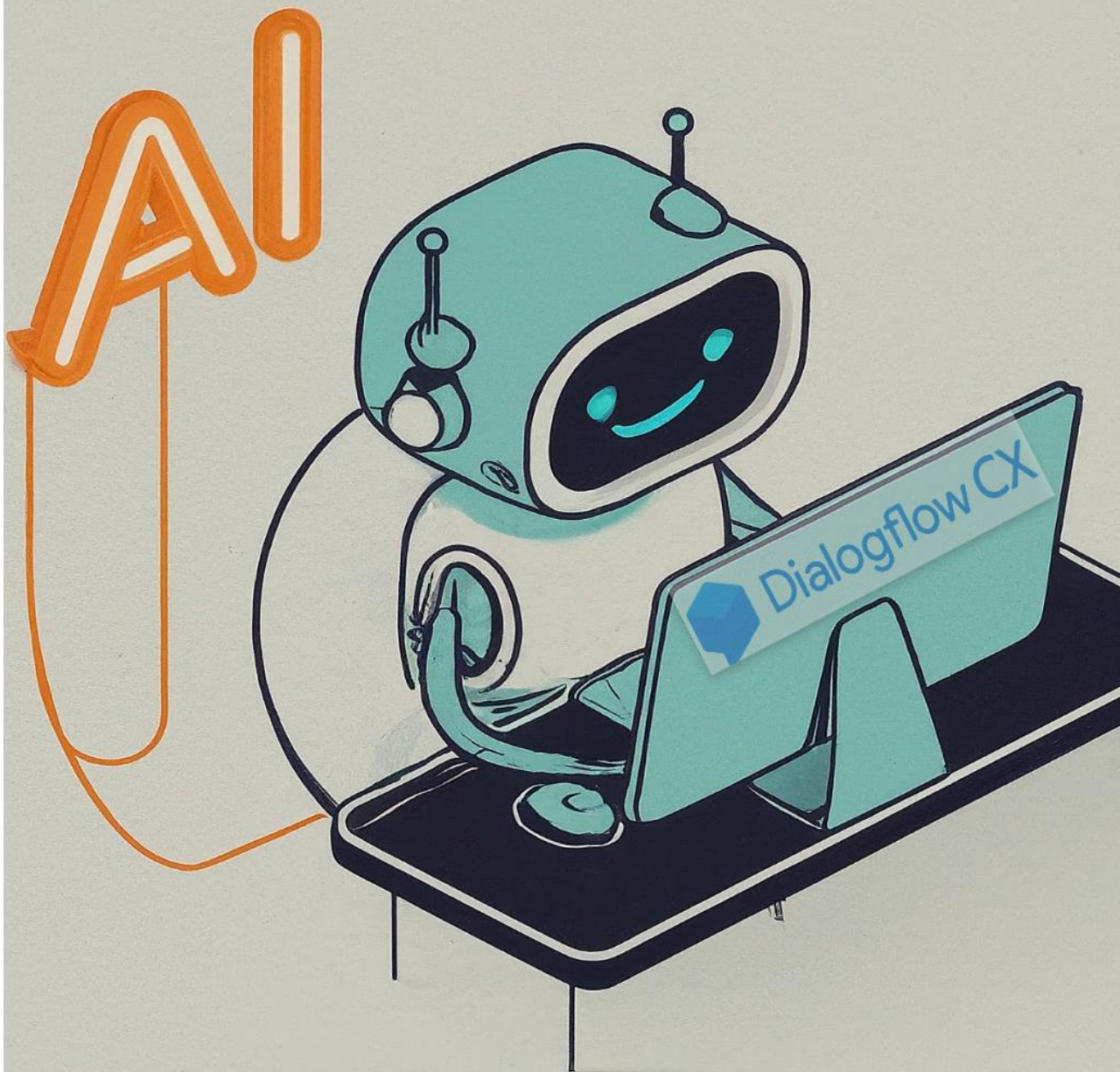
## Kontekst

Pozwoli chatbotowi zachować ciągłość rozmowy.

Przykłady:

- U: „Jakie są godziny otwarcia dziekanatu?”
- Bot: „Dziekanat jest otwarty od 9:00 do 15:00. Czy chcesz znać adres?”
- Użytkownik: „Tak.”





# Proces implementacji chatbota

Implementacja składa się z następujących kroków

- Dodanie nowych intents
- Określenie fraz trenujących
- Dodanie niezbędnych encji
- Określenie odpowiedzi generowanych przez chatbota
- Integracja z platformą Discord



# Najważniejsze funkcje chat-bota

1

Informacja o  
prowadzących  
przedmiot

2

Rekomendacje dla  
studentów

3

Polecanie Politechniki  
Krakowskiej

4

Przyjmowanie zgłoszeń  
o prowadzących

# Etapy integracji Dialogflow z Discordem

Utworzenie  
serwera discord

- Na tym serwerze będzie zarejestrowana aplikacja, która będzie się komunikować z dialogflow

Zarejestrowanie  
aplikacji

- Aplikacja na Discord umożliwia na komunikowanie się z zewnętrznymi usługami oraz wyświetlanie wyników na serwerze

Pobranie tokenu

- W panelu Google Cloud developer trzeba znaleźć nasz projekt dialogflow, a następnie wygenerować token

Zaprogramowanie  
chatbota

- Kod robota ma na celu zestawienie połączenia z dialogflow. Przed tym trzeba mu ustawić odpowiedni token, który wcześniej wygenerowaliśmy



# Integracja z Discordem - kod źródłowy

```
import os
import discord
from google.cloud import dialogflow_v2 as dialogflow
from dotenv import load_dotenv

load_dotenv()

DISCORD_TOKEN = os.getenv('DISCORD_TOKEN')
GOOGLE_APPLICATION_CREDENTIALS = os.getenv('GOOGLE_APPLICATION_CREDENTIALS')

print(f'DISCORD_TOKEN: {DISCORD_TOKEN}')
print(f'GOOGLE_APPLICATION_CREDENTIALS: {GOOGLE_APPLICATION_CREDENTIALS}')
```

Google DialogFlow



Other Chatbots



when the random  
query is asked

```
intents = discord.Intents.default()
intents.message_content = True

client = discord.Client(intents=intents)

def detect_intent_texts(project_id, session_id, text, language_code):
    os.environ["GOOGLE_APPLICATION_CREDENTIALS"] =
GOOGLE_APPLICATION_CREDENTIALS
    session_client = dialogflow.SessionsClient()
    session = session_client.session_path(project_id, session_id)
    text_input = dialogflow.TextInput(text=text, language_code=language_code)
    query_input = dialogflow.QueryInput(text=text_input)
    response = session_client.detect_intent(session=session,
query_input=query_input)
    return response.query_result.fulfillment_text

@client.event
async def on_ready():
    print(f'We have logged in as {client.user}')

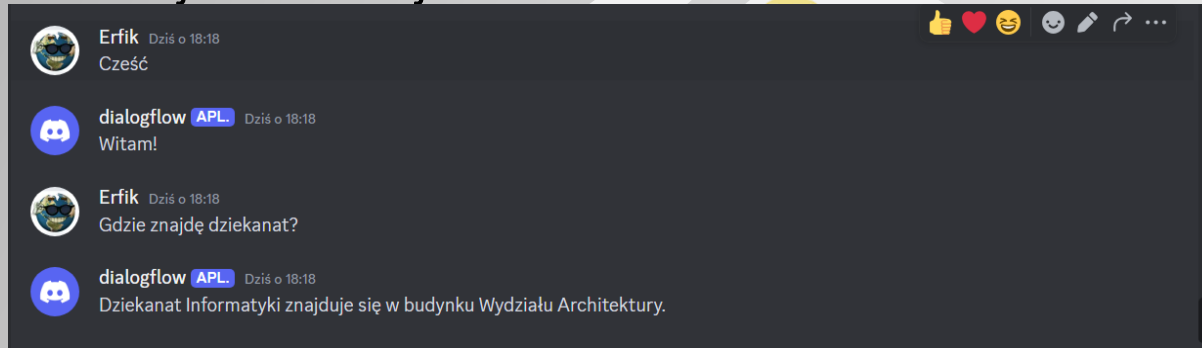
@client.event
async def on_message(message):
    if message.author == client.user:
        return

    project_id = 'bot-weso-y-student-adha'
    session_id = message.author.id
    response_text = detect_intent_texts(project_id, session_id,
message.content, 'pl')
    await message.channel.send(response_text)

client.run(DISCORD_TOKEN)
```

# Przykłady interakcji z chatbotem

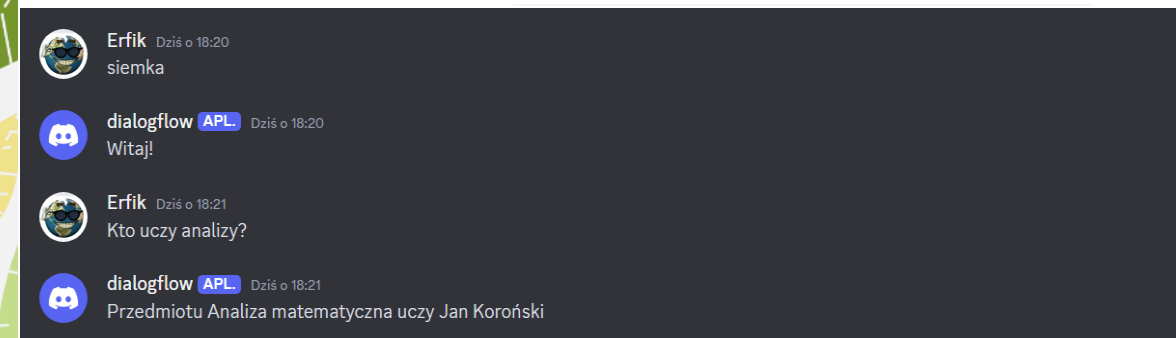
## Informacje o lokalizacji dziekanatu



## Czas wolny



## Pytanie o prowadzącego przedmiot





Dziękujemy!