

PJN projekt praktyczny

Konstrukcja ChatBota w oparciu o Dialogflow.

Autorzy:

Aleksandra Rudy

Arkadiusz Skrzypek

Bartosz Preweda

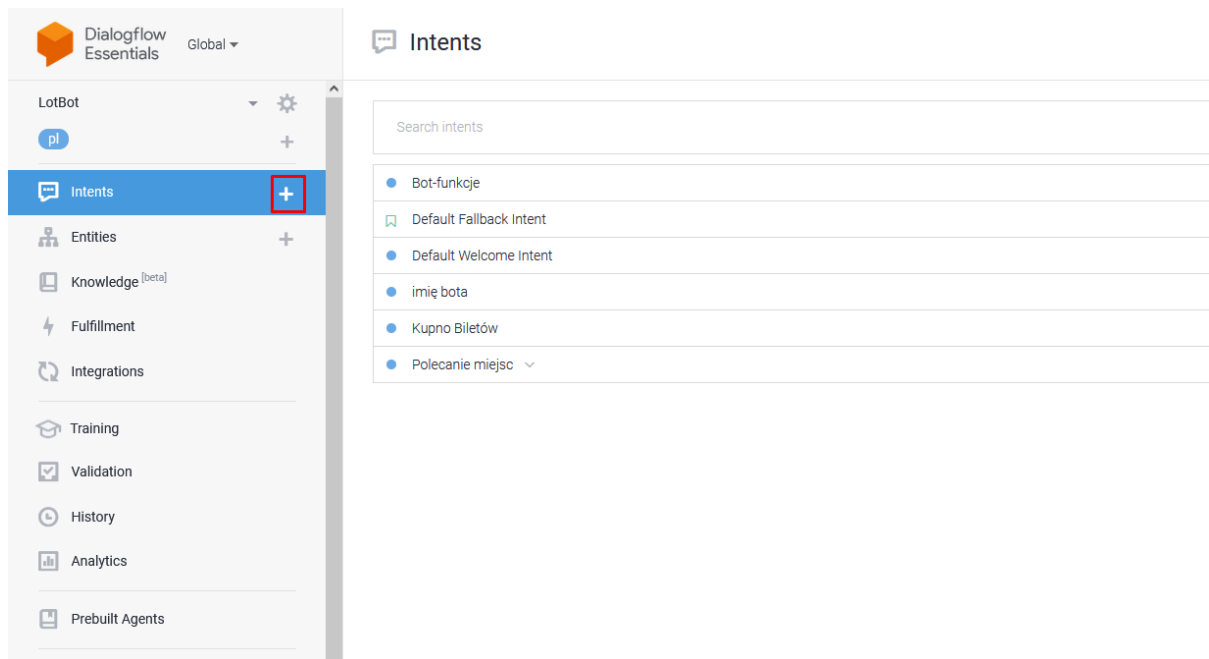
Wstęp

Dialogflow to narzędzie firmy Google do budowania konwersacyjnych aplikacji (jak *chatboty* i *voice boty*), oparte na autorskim rozwiązaniu rozumienia języka naturalnego (ang. Natural Language Understanding, NLU) oraz wykorzystujące doświadczenia i produkty Google z zakresu uczenia maszynowego (ang. Machine Learning, ML) i sztucznej inteligencji (ang. Artificial Intelligence, AI).

Zdecydowaliśmy się na budowę ChatBota, który będzie odpowiedzialny za kupno biletów na lot, ale dodatkowo będzie potrafił polecić użytkownikowi miejsce do którego można polecieć.

Jak działa dialogflow

Gdy mamy utworzony obiekt Bota tzw. "Agenta" musimy dodać do niego "Intents" - czyli jedno-tematyczne konwersacje, które każda z nich będzie miała swoje frazy treningowe oraz ustalone odpowiedzi.



Zacznijmy od prostego zapytania o imię/nazwę bota.

Training phrases ⓘ Search training phrases 🔍 ⤴

⚠️ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression

” kto ty?

” kim jesteś?

” Jak się nazywasz?

Action and parameters ⌵

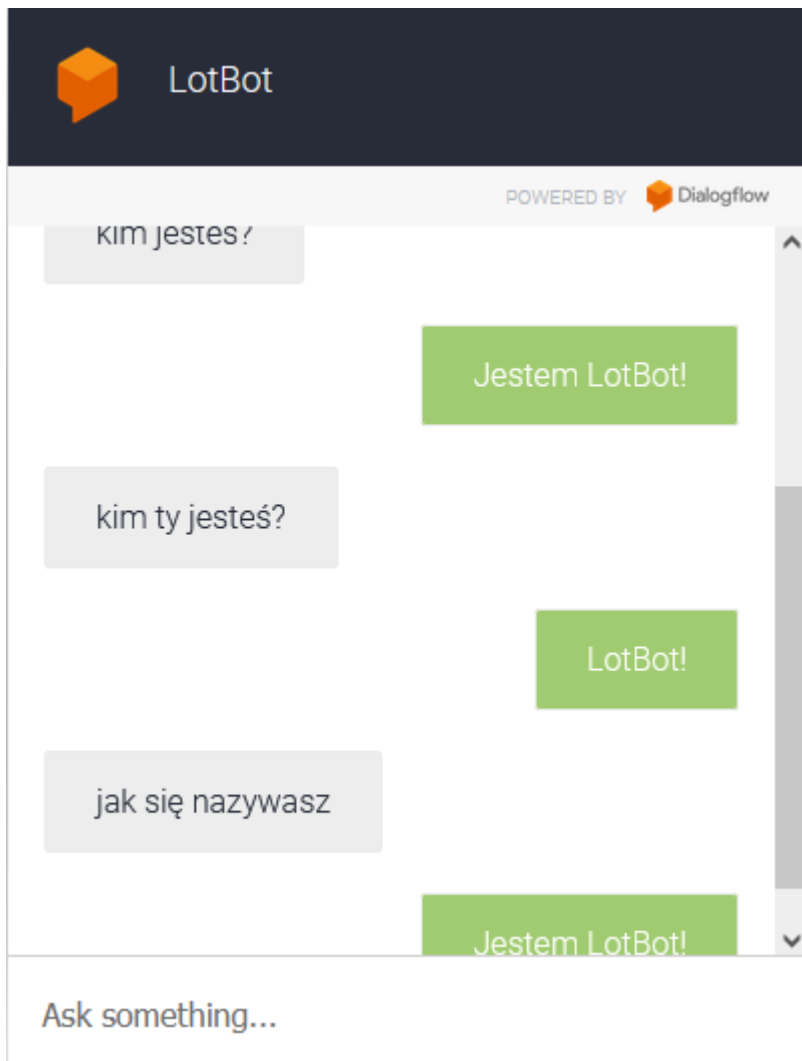
Responses ⓘ ⤴

DEFAULT +

Text Response 🗑

| | |
|---|-------------------------------|
| 1 | LotBot! |
| 2 | Jestem LotBot! |
| 3 | Enter a text response variant |

Wpisujemy kilka fraz treningowych , oraz kilka odpowiedzi. Należy zwrócić uwagę na to, że wyświetla się informacja o tym, że bot sobie sam poradzi z wiadomościami które się nieznacznie różnią. Dzięki temu nasz bot jest w stanie odpowiedzieć nam na te pytania:







Teraz zajmiemy się czymś bardziej skomplikowanym , a dokładnie prostym “systemem” do kupna biletów lotniczych. Dodajemy więc kolejny Intent i zastanawiamy się co musimy wiedzieć aby umożliwić kupno biletów użytkownikowi. Założymy, że nasz prosty system będzie potrzebował tylko ilości osób, miejsce do którego chce polecieć oraz klasę w której chce lecieć (pierwsza/druga/ekologiczna).

Tworzymy więc frazy treningowe i tym razem użyjemy parametrów:
ilość osób,
lokacja(miejsce docelowe lotu)
klasa lotu.


DialogFlow pozwala nam na “nałożenie” parametrów na frazy treningowe co widzimy na poniższym zdjęciu. Zaznaczamy(lub wybieramy interesujące nas parametry, bo dialogFlow automatycznie je wykrywa i nakłada). więc wszędzie tam gdzie chodzi o


ilość osób ustawiamy jako parametr number-integer, miejsce docelowe to location, a klasa lotu to nasz customowy parametr(będzie o tym później).


Training phrases  Search training phrases  




 Template phrases are deprecated and will be ignored in training time. More details [here](#).


When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.


 Add user expression

 Jeden bilet w pierwszej klasie do Korei


 Chce kupić jeden bilet do Niemczech w pierwszej klasie






| PARAMETER NAME | ENTITY | RESOLVED VALUE | |
|----------------|---------------------|----------------|---|
| number-integer | @sys.number-integer | jeden |  |
| location1 | @sys.location | Niemczech |  |
| KlasaLotu | @KlasaLotu | pierwszej |  |

 interesuje mnie kupno 2 biletów do Rzymu w ekologicznej klasie

 Chce kupić bilety dla 4 osób na lot do Francji w drugiej klasie

Następnie musimy ustawić odpowiednie opcje dotyczące naszych parametrów:

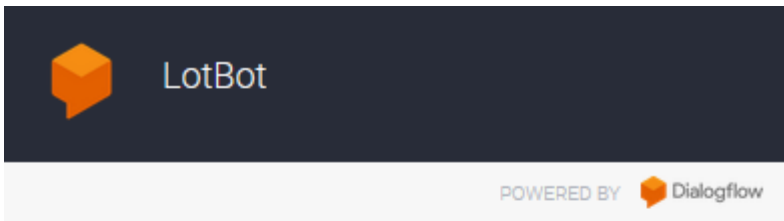
Action and parameters 

| REQUIRED  | PARAMETER NAME  | ENTITY  | VALUE | IS LIST  | PROMPTS  |
|--|--|--|------------------|---|---|
| <input checked="" type="checkbox"/> | number-integer | @sys.number-integer | \$number-integer | <input type="checkbox"/> | Dla ilu osób? [...] |
| <input checked="" type="checkbox"/> | location1 | @sys.location | \$location1 | <input type="checkbox"/> | dokąd? [2] |
| <input checked="" type="checkbox"/> | KlasaLotu | @KlasaLotu | \$KlasaLotu | <input type="checkbox"/> | w jakiej klasie... |
| <input type="checkbox"/> | Enter name | Enter entity | Enter value | <input type="checkbox"/> | — |

[+ New parameter](#)

Głównie co nas tu interesuje, to czy dany parametr jest wymagany do uzyskania odpowiedzi, oraz zakładka “prompts” - która określa odpowiedź ChatBota gdy zapytanie użytkownika nie będzie posiadało danego parametru.

Na przykład gdy użytkownik będzie chciał zarezerwować lot, ale nie poda dla ilu osób:



Chciałbym zarezerwować lot do Argentyny w pierwszej klasie

Dla ilu osób?

czterech

Kupiono 4 biletów w pierwszej klasie do Argentyna - dalsze informacje otrzymasz na maila

Ask something...

Jak można zauważyć, możemy stworzyć odpowiedź która będzie zawierała parametry

Text Response

1 Kupiono Snumber-integer biletów w SKlasaLotu klasie do Slocation1 - dalsze informacje otrzymasz na maila

Wracając do klasy lotu. Nie znajdziemy w bazie DialogFlow parametru typu klasa lotu, musimy sami go zrobić. Dodajemy więc Entity:

| Entity | Value |
|--------------|---------------------------|
| pierwszej | pierwsza, pierwszej, 1 |
| drugiej | druga, drugiej, 2 |
| ekologicznej | ekologiczna, ekologicznej |

Click here to edit entry


+ Add a row


Dodajemy nazwę naszego Entity, oraz wartości jakie może przyjmować, oraz ich synonimy. W naszym przypadku to jest klasa pierwsza/druga/ekologiczna.

Ostatnią funkcjonalnością jest polecenie użytkownikowi miejsca lotu.

Tworzymy Intent polecenia miejsc, gdzie pytamy użytkownika jaki klimat preferuje:

Polecanie miejsc

Events 

Training phrases 



Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list of expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression

” jakieś propozycje gdzie mogę polecieć?

” Czy możesz mi polecić jakieś miejsce lotu?

Action and parameters

Responses 

DEFAULT +

Text Response

1 Preferujesz ciepły czy zimny klimat?

2 Enter a text response variant

Następnie dodajemy dwa follow-up Intenty - czyli konwersacje które trzymają kontekst poprzedniej wypowiedzi. Tworzymy jeden intent dla klimatu ciepłego i jeden dla klimatu zimnego I tam również dodajemy frazy treningowe.

● Polecanie miejsc ^

● ↳ Polecanie miejsc - zimne

● ↳ Polecanie miejsc - ciepłe

Ostatecznie jak wygląda przykładowa rozmowa:

Hej jak się nazywasz?

LotBot!

co potrafisz?

Zajmuje się polecaniem miejsc do których można polecieć, oraz kupnem/rezerwacją biletów na lot.

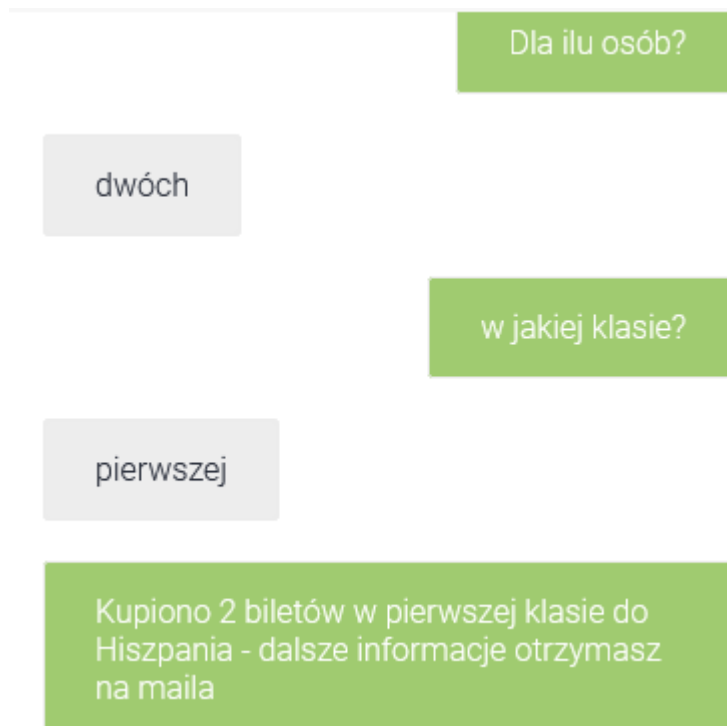
w takim razie co mi możesz polecić?

Preferujesz ciepły czy zimny klimat?

ciepły

Może zawsze gorąca Hiszpania?

może być, chce kupić bilety do Hiszpanii



Podsumowanie

Rozwiązanie DialogFlow od google, pozwala na proste i przejrzyste robienie ChatBota. Wielką zaletą jest przejrzysty graficzny interfejs, oraz dobrze zrobiona dokumentacja. Dodatkowo użytkownik nie musi umieć kodować, ponieważ wszystko robi się za pomocą graficznego interfejsu.

ChatBota można przetestować tutaj:

<https://bot.dialogflow.com/4b44b80b-0091-4d6d-abfa-4492ed7fe06e>

Bibliografia

<https://cloud.google.com/dialogflow/docs>