



# Bot dla call-center przy użyciu AWS Cloud i Amazon Lex

P.S.  
P.P.



## Czym jest AWS

AWS to jedna z największych publicznych chmur obliczeniowych. Dostarcza wielu różnych usług, od zasobów obliczeniowych do rozwiązań sztucznej inteligencji.



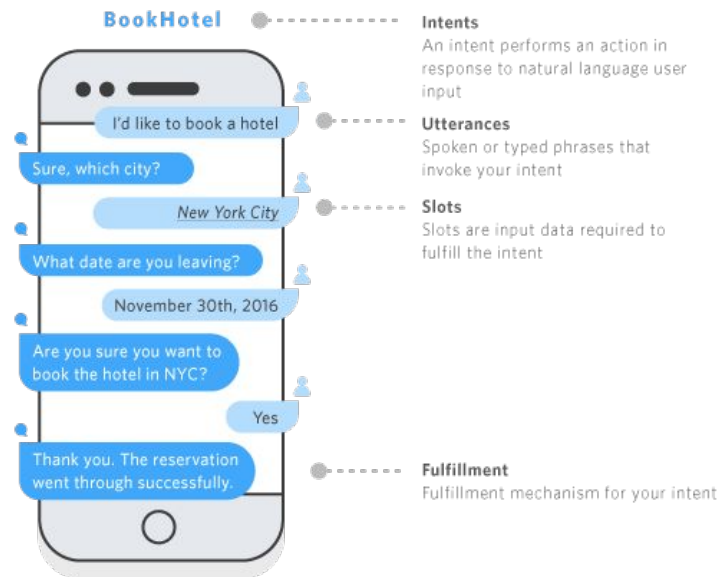
# Czym jest Amazon Lex

Amazon Lex to usługa AWS służąca do budowania interfejsów konwersacyjnych dla aplikacji wykorzystujących głos i tekst.



# Funkcje

Amazon Lex korzysta z tej samej technologii co Alexa. Umożliwia przetwarzanie języka naturalnego i mowy. Pozwala na integrację z zewnętrznymi systemami w celu automatyzacji pobierania i zapisywania informacji. Może zostać wdrożone na platformach czatowych, urządzeniach mobilnych, bądź urządzeniach IoT. Boty mogą w różnych językach, niestety Amazon Lex nie wspiera języka polskiego.



---

## Amazon Lex w rozwiązaniach call-center

Operatorzy call-center mogą korzystać z usługi Amazon Lex, aby zapewnić naturalne wrażenia z konwersacji, optymalizując modele świadczenia usług i udostępniając klientom nowe technologie komunikacji i mediów.

Gotowe chatboty umożliwiają automatyzację interakcji z obsługą klienta, takich jak aktywacja usługi, opłacanie rachunków, wyjaśnienie ofert i zarządzanie instalacjami.





## Zasada działania

Amazon Lex pozwala na tworzenie 2 typów wiadomości:

- prompt - pytanie na, które bot oczekuje odpowiedzi lub nie rozumie kontekstu wypowiedzi
- statement - wiadomość informacyjna, bot nie oczekuje odpowiedzi

Wiadomości mogą zawierać dynamiczne informacje poprzez użycie slotów i atrybutów. Na przykład:

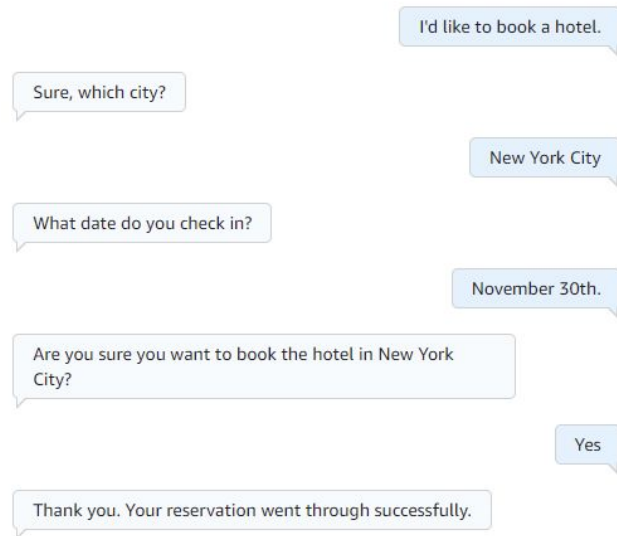
```
"Hey [FirstName], your {PizzaTopping} pizza will arrive in [DeliveryTime] minutes."
```

```
"Hey John, your cheese pizza will arrive in 30 minutes."
```

# Przykładowy bot

Przedstawiony zostanie przykład implementacji bota, służącego do rezerwacji pokoju hotelowego. W pierwszej kolejności musimy przedstawić swoje intencje, czyli chęć zabookowania hotelu. W przypadku rozpoznania intencji, bot poprosi nas o podanie szczegółowych informacji odnośnie rezerwacji, takich jak: miejsce oraz data.

Następnie bot upewni się, że na pewno chcemy dokonać takiej akcji (rezerwacji). Jeżeli zostanie wykryte finalne potwierdzenie bot podziękuję nam oraz zakończy konwersację powiadamiając nas o sukcesie lub porażce (w naszym przypadku bookowania).





# Wywołanie akcji

Konfiguracje intencji należy zacząć od podania przykładowych fraz, które może wpisać użytkownik czat bota. Amazon Lex na podstawie podanych zdań automatycznie stworzy model pozwalający najlepiej przewidywać intencje użytkownika.

### Sample utterances (3) [Info](#)

Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the samples. The priority order of the sample utterances is not used to determine intent classification output.

- Book a hotel
- I want a make hotel reservations
- Book a **{Nights}** night stay in **{Location}**

Maximum 250 characters.



# Przebieg konwersacji

Przebieg konwersacji można ustawić przez dodanie slotów. Ich celem jest pozyskanie informacji od użytkownika. Sloty mają swoją nazwę, typ informacji, która jest oczekiwana (np. lokalizacja, data, liczby, itp.) i wiadomości dla użytkownika.

The screenshot shows the configuration page for slots in a bot framework. At the top, there is a section titled "Slots (4) - optional" with an "Info" link and an "Add slot" button. Below this is a search bar labeled "Filter". The main area contains a list of four configured slots, each with a vertical ellipsis on the left and a close button (X) on the right. Each slot entry includes a prompt for the slot, a sample message, and the slot type.

Slot Name	Message	Slot Type
Prompt for slot: Location	Message: What city will you be staying in?	AMAZON.City
Prompt for slot: CheckInDate	Message: What day do you want to check in?	AMAZON.Date
Prompt for slot: Nights	Message: How many nights will you be staying?	AMAZON.Number
Prompt for slot: RoomType	Message: What type of room would you like, queen, ki...	RoomTypeValues



# Dodatkowe potwierdzenie

Amazon Lex pozwala na skonfigurowanie dodatkowego potwierdzenia przed wykonaniem akcji. Umożliwia to użytkownikowi anulowanie akcji gdyby się rozmyślił lub zmienił zdanie. Ten krok jest opcjonalny.

### Confirmation prompts and decline responses [Info](#)

Prompts help to clarify whether the user wants to fulfill the intent or cancel it. Active

▼ Prompts to confirm the intent <i>Message: Okay, I have you down for a {Nights} night s...</i>	Responses sent when the user declines the intent <i>Message: Okay, I have cancelled your reservation in pr...</i>
<p><b>Confirmation prompt</b> What will the bot say to prompt the user to confirm this intent.</p> <input type="text" value="Okay, I have you down for a {Nights} night stay in {Location} starting {CheckInDate}. Shall I book the reservation?"/>	
<p><b>Decline response</b> What will the bot say if the user says NO to the confirmation prompt.</p> <input type="text" value="Okay, I have cancelled your reservation in progress."/>	
<p><b>Advanced options</b></p> <p>Configure confirmation prompts and decline responses.</p>	



# Status

W tym kroku jesteśmy zobligowani do uzupełniania informacji związanych z powiadomieniem użytkownika o statusie, wykonywanej przez bota akcji. Wypełniamy co powinno stać się w przypadku sukcesu i porażki oraz sentencję zamykającą niezależną od typu statusu.

**Fulfillment** [Info](#)  
Run a lambda function to fulfill the intent and inform users of the status when it's complete.

<p>▶ On successful fulfillment Message: -</p>	<p>In case of failure Message: -</p>
---	--

**Closing response** [Info](#)  Active  
You can define the response when closing the intent.

▶ Response sent to the user after the intent is fulfilled  
Message: -

# Podgląd przykładowej konwersacji

Po skonfigurowaniu konwersacji, Amazon Lex pokaże podsumowanie oraz przykładowy przebieg konwersacji.

The screenshot displays a sample conversation flow for booking a hotel, divided into four stages:

- 1 of 4:** Initial request - sample utterance: "Book a hotel". Prompt for more information - slot: "What city will you be staying in?". Capture information - slot value: "<city>". Prompt for more information - slot: "What day do you want to check in?".
- 2 of 4:** Capture information - slot value: "<date>". Prompt for more information - slot: "How many nights will you be staying?". Capture information - slot value: "<number>". Prompt for more information - slot: "What type of room would you like, queen, king or deluxe?".
- 3 of 4:** Capture information - slot value: "queen". Confirm intent - confirmation prompt: "Okay, I have you down for a {Nights} night stay in {Location} starting {CheckInDate}. Shall I book the reservation?". Notify that fulfillment started - fulfillment updates: "e.g. Thank you for the information. We have started the process." and "e.g. We are still working on it, please wait a moment".
- 4 of 4:** Fulfillment completed successfully - success response: "e.g. The process is completed, thank you.". Fulfillment failed to complete - failure response: "e.g. Sorry, something went wrong. We will get back to you.". Fulfillment timed out - timeout response: "e.g. Sorry, we are having issues with the process. We will get back to you.". Send final response - closing response: "Thank you, have a good day".



**Dziękujemy za uwagę**