



Building Chatbots with Amazon Lex V2

Introductions



Jeremy Cook

Content Lead Architect

Cloud Academy DevOps Team Lead

Background in DevOps, Cloud, Security, Kubernetes, Machine Learning. Software Engineering, various languages (Python, C#, Go, Java, Kotlin, NodeJS) and related frameworks.

Professional certifications for AWS, GCP, and Kubernetes.



<https://www.linkedin.com/in/jeremycook123/>



support@cloudacademy.com



Cloud Training That Drives Digital Transformation

<https://cloudacademy.com>

cloud academy A COMPANY

Platform Training Library Pricing Resources Contact Search in our library... My Dashboard

Build hands-on technical skills. Drive measurable results.

Assess, develop, and validate tech skills with structured training plans, skill validation in live environments, and built-in accountability.

FOR ENTERPRISE FOR PERSONAL

Try Demo Start Free Trial

Training Plan: Acme Inc. Solution Architect

DUE DATE April 22

Skills Improvement

Expected skill growth: 80%

EXAM SCORE 85/100 +2.4%

PROGRESS 77% On track

January (Start Date) February Today April (Due Date)

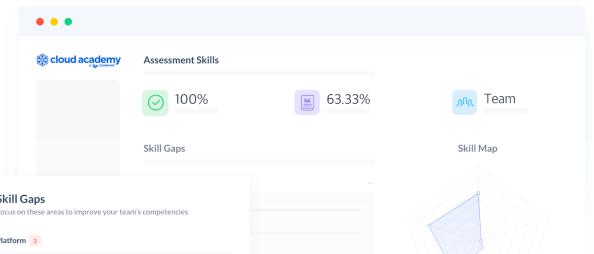
TRUSTED BY: SAS Cognizant NASA BankUnited SAP

ASSESS

Accurately pinpoint baseline skills

Test competence across multiple cloud platforms and technologies and track skill improvement.

- Test practical, hands-on tech skills
- Streamline the assessment process with automated reminders





Amazon Lex - In-depth Review X +

https://cloudacademy.com/course/amazon-lex-chatbots/amazon-lex-in-depth-review-and-detail-1

Dashboard / Training Library / Amazon Web Services / Courses / Working with Amazon Lex - Chatbots

Amazon Lex - In-depth Review and Detail

Send to your phone

Contents

Learning Path

AMAZON LEX - DEEP DIVE

- 1 Amazon Lex - Introduction 2m 10s
- 2 Amazon Lex - In-depth Review and Detail 13m 20s

CREATING A LEX BOT

- 3 Overview / Setting Up The Environment 6m 1s
- 4 Create Lambda Fulfilment Function 11m 34s
- 5 Create Lex Bot with Intents and Slots 8m 17s
- 6 Running the Lex Bot with Amazon Polly 6m 50s

ChatBots

Amazon Lex

CloudAcademy

OVERVIEW TRANSCRIPT

Difficulty Duration Students

Intermediate 49m 861

Report an issue

Invite Guest

Feedback





[GitHub - cloudacademy/aws-lexv2-chatbot](https://github.com/cloudacademy/aws-lexv2-chatbot)

cloudacademy / aws-lexv2-chatbot Public

Code Issues Pull requests Actions Projects Wiki Security Insights

Go to file Code

main 1 branch 0 tags

jeremycook123 added retries for lambda creation b3de7e7 21 hours ago 21 commits

- doc/images updates 2 days ago
- lambda updates 2 days ago
- lex updates 2 days ago
- .gitignore updates 2 days ago
- README.md updates 2 days ago
- install.sh added retries for lambda creation 21 hours ago

README.md

Amazon Lex Movie Recommendations Chatbot

Introduction

A simple [Amazon Lex](#) Chatbot that is able to provide movie recommendations for a given movie genre. This implementation is performed using [Amazon Lex V2](#).

Integration

Movie recommendation data is provided by The Movie Database ([TMDB](#)).

TMDB Movies TV Shows People More + EN 🔔 🔍

About

AWS Lex v2 Chatbot

aws devops chatbot tmdb lex cloudacademy

Readme

Releases

No releases published

Packages

No packages published

Languages

Python 54.6% Shell 45.4%



What Are Chatbots

Chatbot – Simulated Conversations



chatbot

/'tʃatbot/

noun

plural noun: **chatbots**

a computer program designed to simulate conversation with human users, especially over the internet.

"chatbots often treat conversations like they're a game of tennis: talk, reply, talk, reply"

Origin

ENGLISH

chat

ENGLISH

bot

chatbot

1990s

1990s: from **chat¹** + **bot¹**.

Translate chatbots to

Choose language ▾



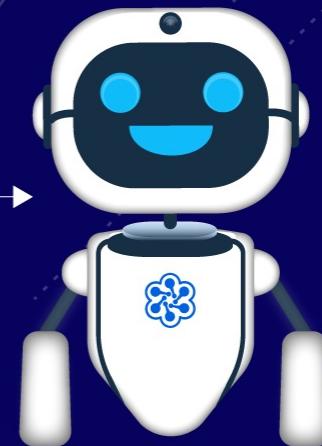
I have a
technical query...



Can you help
me find...

?

?



Chatbot Use Cases

Real-life uses of Chatbots

Chatbot Channels

Both Text and Voice Based



Business Drivers to Adopt Chatbots

Motivations for Implementing a Chatbot

- Immediate Response
- 24/7 Availability
- No queues
- Workforce Cost (none)
- Workforce Focused on Business Value Add Activities



Amazon Lex

Chatbot and Conversational AI Managed Service

- Chatbots
- Conversational AI
- Managed Service
- Cost Effective
- Omnichannel
- No Upfront Costs
- NLP Models

The screenshot shows the AWS Amazon Lex Overview page. At the top, there's a navigation bar with links like Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, and a search bar. A prominent orange button on the right says "Sign In to the Console". Below the navigation, there's a banner for "AWS Solutions QnABot". The main heading is "Amazon Lex" with the subtitle "Build chatbots with conversational AI". A large orange button says "Get Started with Amazon Lex". To the right, a callout box highlights "10,000 text and 5,000 speech requests free for 12 months with the AWS Free Tier". Below this, four dark boxes describe features: "Easily add AI to your applications that understands intent, maintains context, and automates simple tasks across many languages.", "Deploy omnichannel conversational AI in one click, without worrying about hardware and infrastructure.", "Connect seamlessly with other AWS services to query data, execute business logic, monitor performance, and more.", and "Pay only for speech and text requests with no upfront costs or minimum fees."

How it works

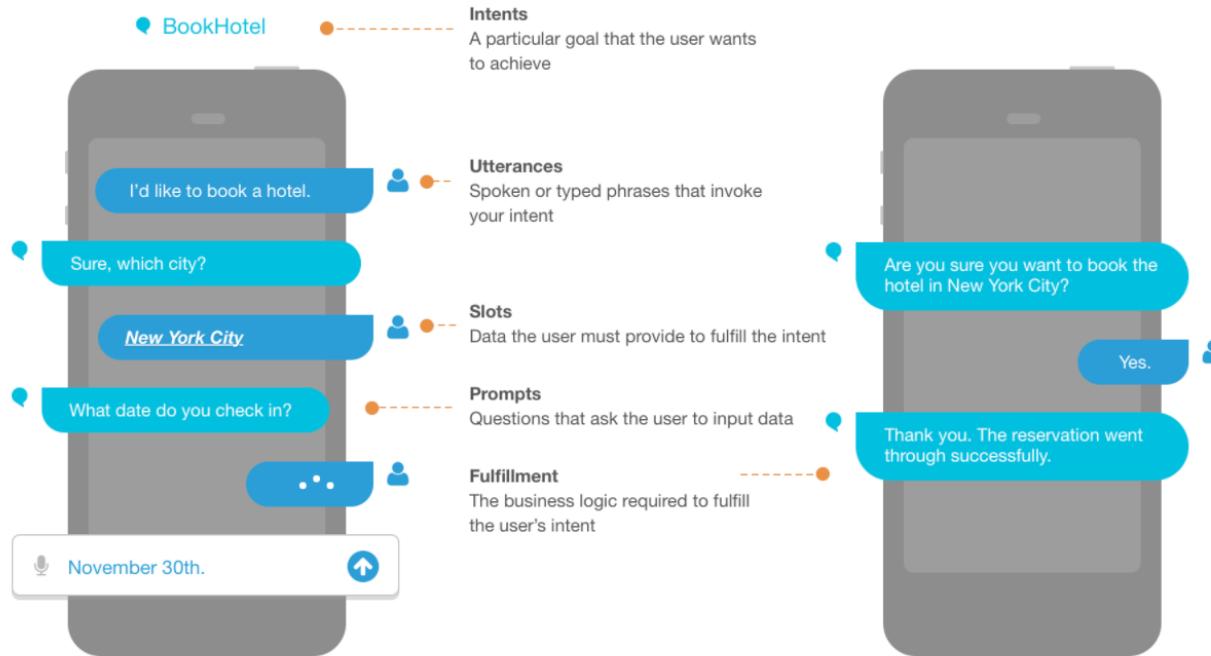
Amazon Lex is a fully managed artificial intelligence (AI) service with advanced natural language models for building conversational interfaces into applications.



Amazon Lex

Understanding the Terminology

- Intents
- Utterances
- Slots
- Prompts
- Fulfillment





Amazon Lex Documentation

https://docs.aws.amazon.com/lex/index.html

aws English ▾ Sign In to the Console

AWS > Documentation > Amazon Lex Feedback Preferences ⓘ

Amazon Lex Documentation

Amazon Lex is an AWS service for building conversational interfaces into applications using voice and text. With Amazon Lex, the same deep learning engine that powers Amazon Alexa is now available to any developer, enabling you to build sophisticated, natural language chatbots into your new and existing applications. Amazon Lex provides the deep functionality and flexibility of natural language understanding (NLU) and automatic speech recognition (ASR) to enable you to build highly engaging user experiences with lifelike, conversational interactions and create new categories of products.

V2 Developer Guide
Provides a conceptual overview of Amazon Lex V2. Includes detailed instructions for using various features, such as streaming conversations, and provides a complete reference for the V2 APIs.
[HTML](#) | [PDF](#) | [GitHub](#)

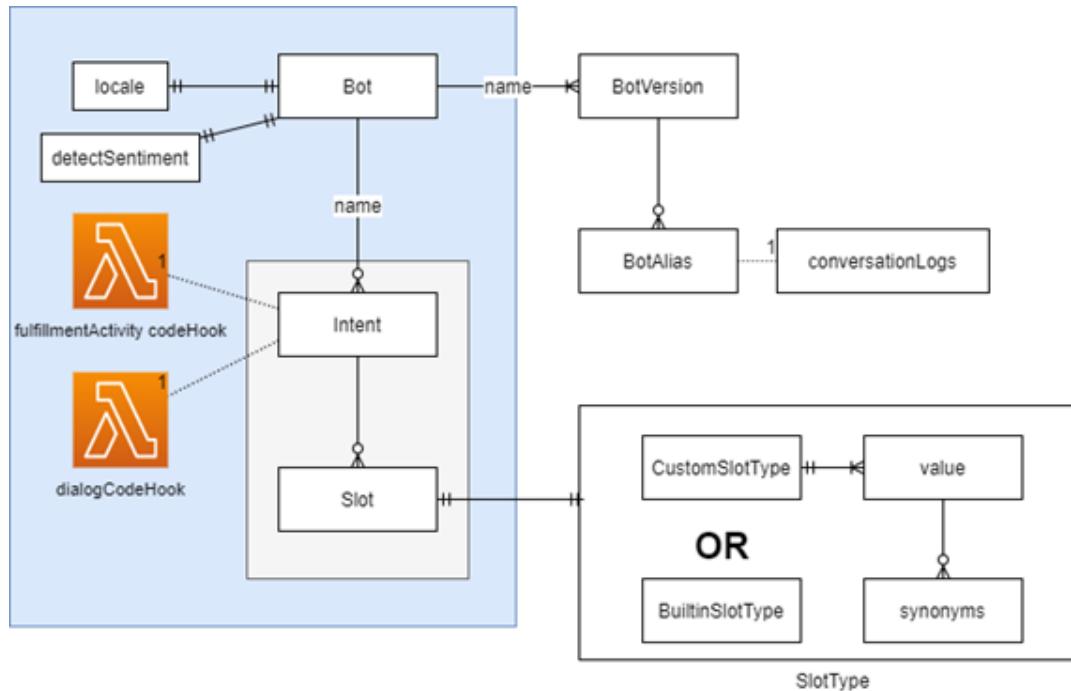
V1 Developer Guide
Provides a conceptual overview of Amazon Lex V1, includes detailed instructions for using the various features, and provides a complete V1 API reference for developers.
[HTML](#) | [PDF](#) | [Kindle](#) | [GitHub](#)



Amazon Lex V1

Lex Entity Model

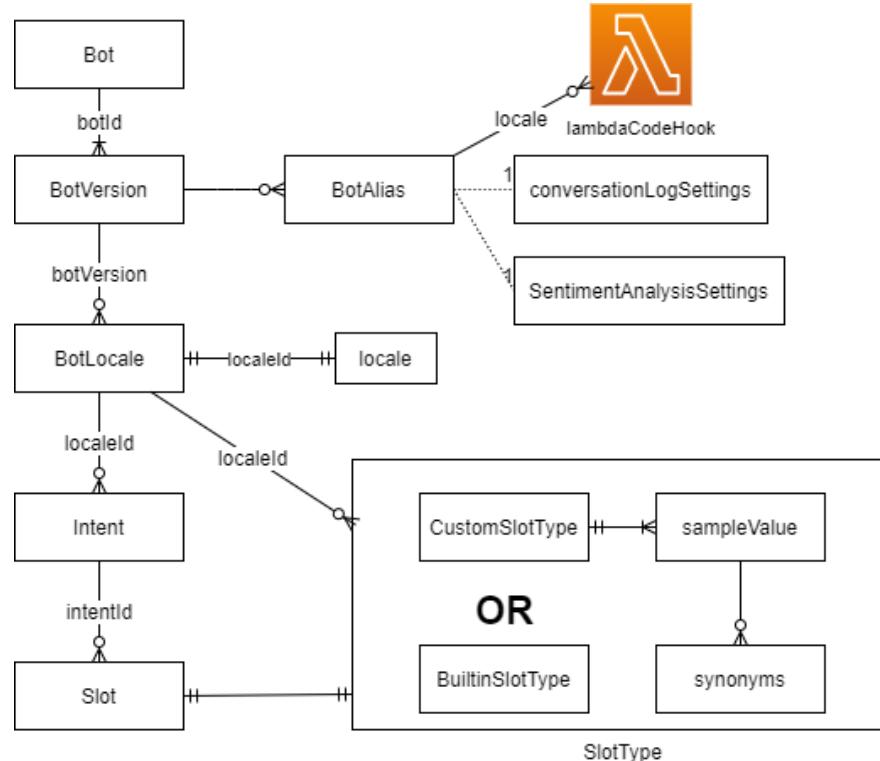
- Bot
- BotVersion
- BotAlias
- Intent
- Slot
- CustomSlotType
- BuiltinSlotType
- CodeHook



Amazon Lex V2

Lex Entity Model

- Bot
- BotVersion
- BotAlias
- Intent
- Slot
- CustomSlotType
- BuiltinSlotType
- CodeHook

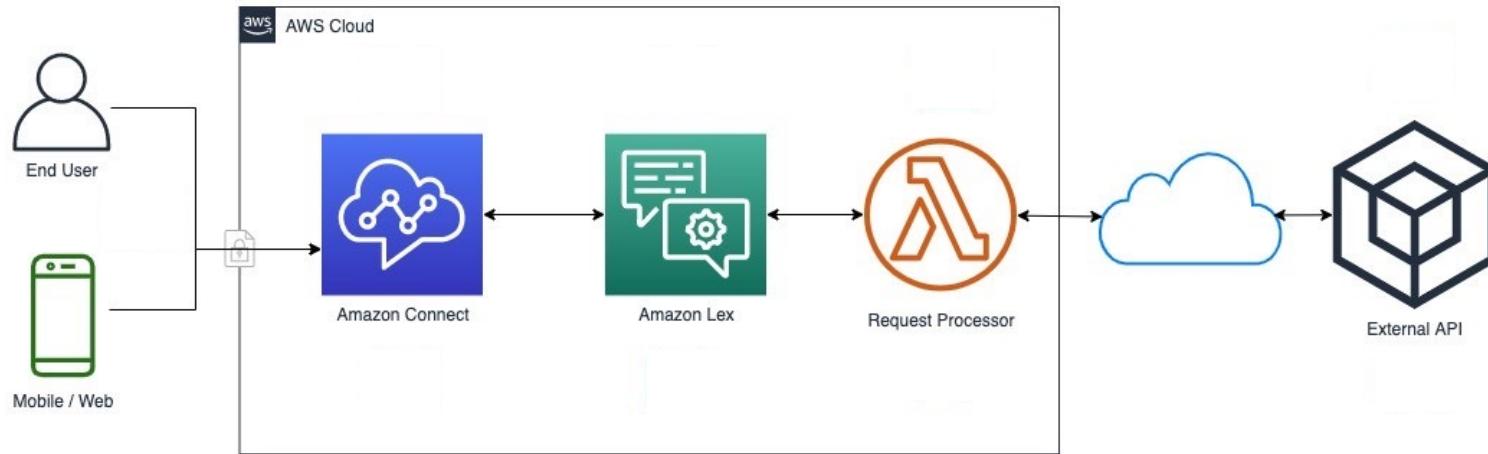


source: <https://www.linkedin.com/pulse/quick-guide-understand-design-difference-between-lex-wong/>



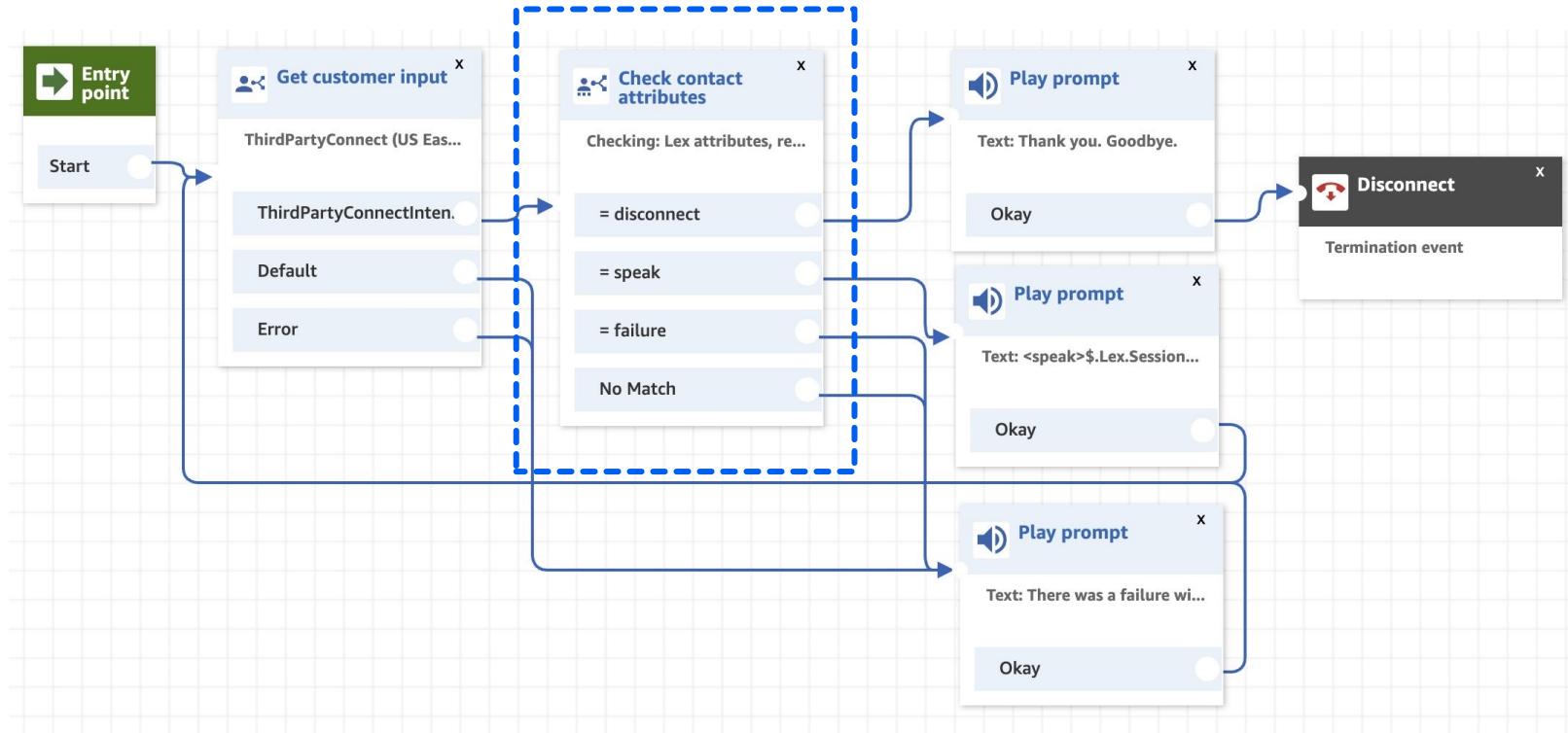
Integration

Seamless Integration with other AWS Services



Amazon Connect

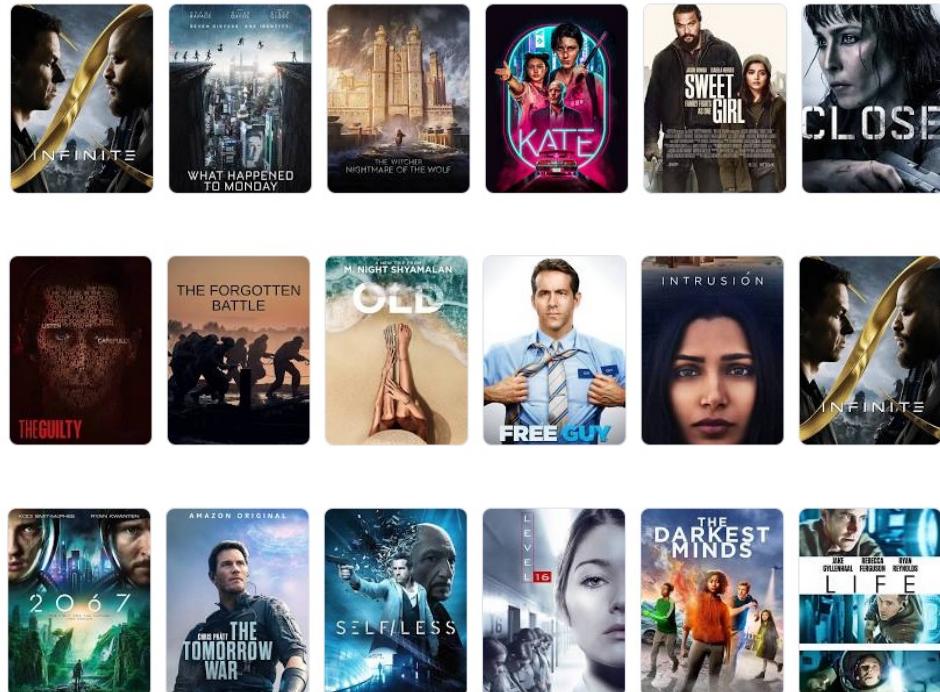
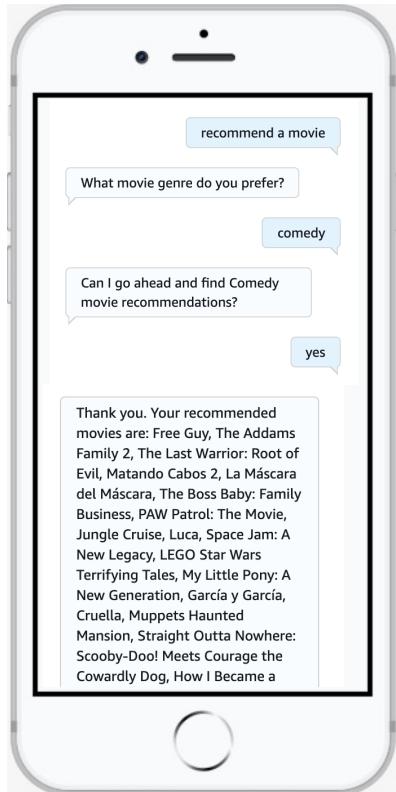
Call Centre Integration



Let's Build a Chatbot

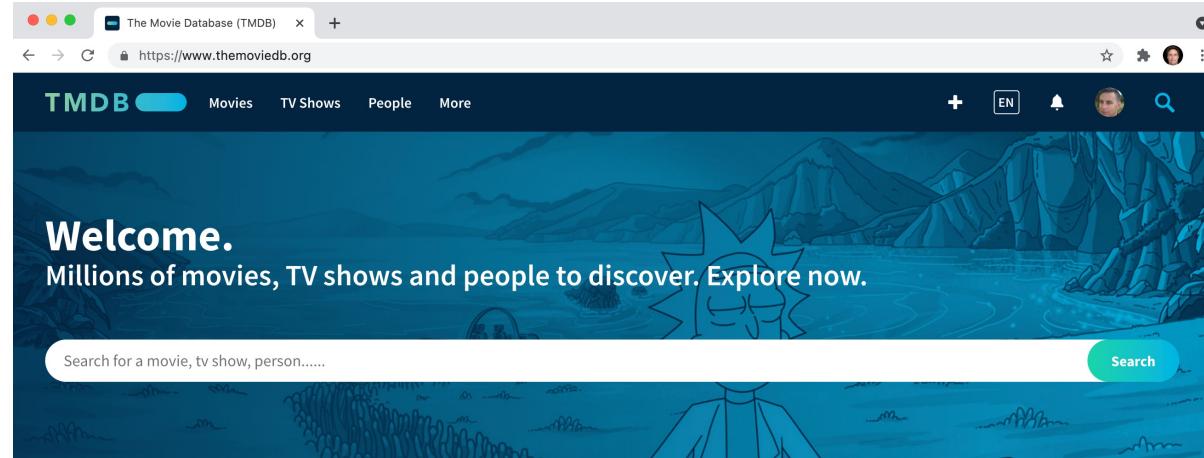
Let's Build a Chatbot

Movie Recommendations



The Movie Database

Online Community Curated Movie Database



What's Popular

Streaming

On TV

For Rent

In Theatres



The Five Juanas
06 Oct 2021



Grey's Anatomy
27 Mar 2005



Madres
08 Oct 2021



You
09 Sep 2018



El Juego de las Llaves
16 Aug 2019



The Flash
07 Oct 2014



My Little Pony: A New Generation
23 Sep 2021



Luca
17 Jun 2021



The Movie Database

Online Community Curated Movie Database

Stats

We all love them. Here's a few that we find interesting.

696,511
Movies

119,589
TV Shows

185,004
TV Seasons

2,861,755
TV Episodes

2,241,203
People

2,971,107
Images

369,134
Edits Last Week

[CONTACT TMDB](#)



Hi cloudacademy!

THE BASICS

About TMDB
Contact Us
Support Forums
API
System Status

GET INVOLVED

Contribution Bible
3rd Party Applications
Add New Movie
Add New TV Show

COMMUNITY

Guidelines
Discussions
Leaderboard
Twitter

LEGAL

Terms of Use
API Terms of Use
Privacy Policy



The Movie Database

The Movie Database API

The screenshot shows a browser window displaying the official documentation for The Movie Database API. The URL in the address bar is <https://developers.themoviedb.org/3/discover/movie-discover>. The page has a dark blue header with the 'THE MOVIE DB' logo and navigation links for 'OAS', 'RAML', and 'Support'. On the left, a sidebar lists various API endpoints: ACCOUNT, AUTHENTICATION, CERTIFICATIONS, CHANGES, COLLECTIONS, COMPANIES, CONFIGURATION, CREDITS, DISCOVER, FIND, GENRES, GUEST SESSIONS, KEYWORDS, LISTS, MOVIES, NETWORKS, TRENDING, PEOPLE, and REVIEWS. Under the 'DISCOVER' section, there is a 'Movie Discover' entry with a 'GET /discover/movie' method. The description explains that this endpoint allows discovering movies by different types of data like average rating, number of votes, genres and certifications. It also supports sort options and filtering by certification and certification_lte with certification_country. A note specifies that if a region parameter is used, the regional release date will be used instead of the primary release date. It also mentions that filters support comma (,) or pipe (|) separated values, where commas are ANDed and pipes are ORed. Examples of discover usage are provided. Below this, a 'Recent Changes' table shows two entries:

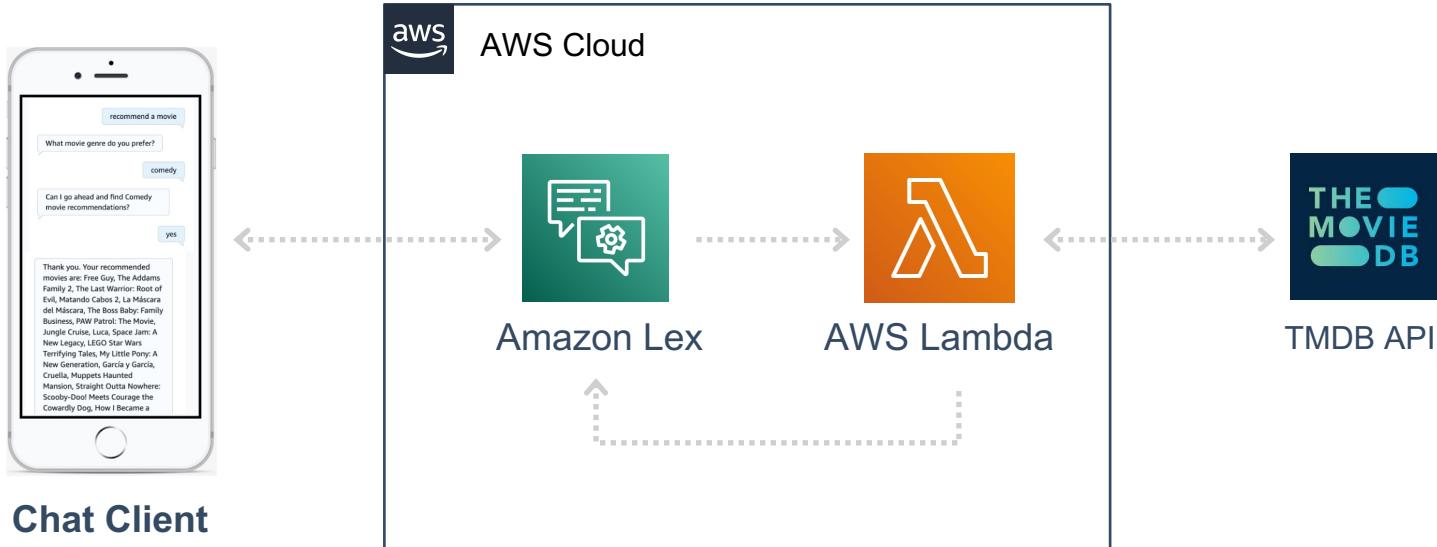
Date	Change
April 13, 2021	A new <code>with_watch_monetization_types</code> filter is available to use with <code>watch_region</code> .
January 2, 2021	A new set of filters are available for watch provider filtering. Check out <code>with_watch_providers</code> and <code>watch_region</code> .

At the bottom, there are 'Definition' and 'Try it out' buttons.



Chatbot Architecture

Movie Recommendations



Movie API Lambda Function

Lex Fulfillment Lambda Function



A screenshot of the AWS Lambda console showing the "MovieAPI" function. The left sidebar shows the navigation menu with "MovieAPI" selected. The main area displays the function overview, including the function name "MovieAPI", a "Layers" section (0 layers), and buttons for "+ Add trigger" and "+ Add destination". Below this is a tabs bar with "Code" (selected), Test, Monitor, Configuration, Aliases, and Versions. The "Code source" tab is open, showing the code editor with the following Python code:

```
File Edit Find View Go Tools Window Test Deploy Changes deployed
Go to Anything (% P)
lambda_function Execution results Preferences +
1 import os
2 from tmdbv3api import TMDb
3 from tmdbv3api import Movie
4 from tmdbv3api import Discover
5
```

The browser address bar shows the URL: https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/MovieAPI?newFunction=true&tab=code

LAMBDA FUNCTION

Perform Query:
TMDB (v3) Movie
API

```
import os
from tmdbv3api import TMDb, Discover

tmdb = TMDb()
tmdb.api_key = os.getenv('APIKEY')

GENRES = {
    "action": 28,
    "adventure": 12,
    "animation": 16,
    "comedy": 35,
    "crime": 80,
    "documentary": 99,
    "drama": 18,
    "family": 10751,
    "Fantasy": 14,
    "history": 36,
    "horror": 27,
    "music": 10402,
    "mystery": 9648,
    "romance": 10749,
    "science fiction": 878,
    "thriller": 53,
```

tmdbv3api 1.7.6

A lightweight Python library for The Movie Database (TMDb) API.

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Statistics

Github statistics:

- Stars: 94
- Forks: 36

Install

```
$ pip install tmdbv3api
```



LAMBDA FUNCTION

Perform Query:
TMDB (v3) Movie
API

Genre Filtered
Search

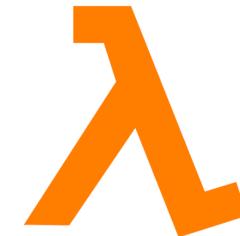
```
def MovieRecommendation(intent_request):
    session_attributes = get_session_attributes(intent_request)
    slots = get_slots(intent_request)
    genre = get_slot(intent_request, 'genre')

    discover = Discover()
    movies = discover.discover_movies({
        'primary_release_date.gte': '2019-01-01',
        'primary_release_date.lte': '2021-12-01',
        'with_genres': GENRES.get(genre.lower())
    })

    recs = ""
    for movie in movies:
        recs = recs + f" {movie.title}, "

    text = f"Thank you. Your recommended movies are: {recs}"
    message = {
        'contentType': 'PlainText',
        'content': text
    }

    fulfillment_state = "Fulfilled"
    return close(intent_request, session_attributes, fulfillment_state, message)
```

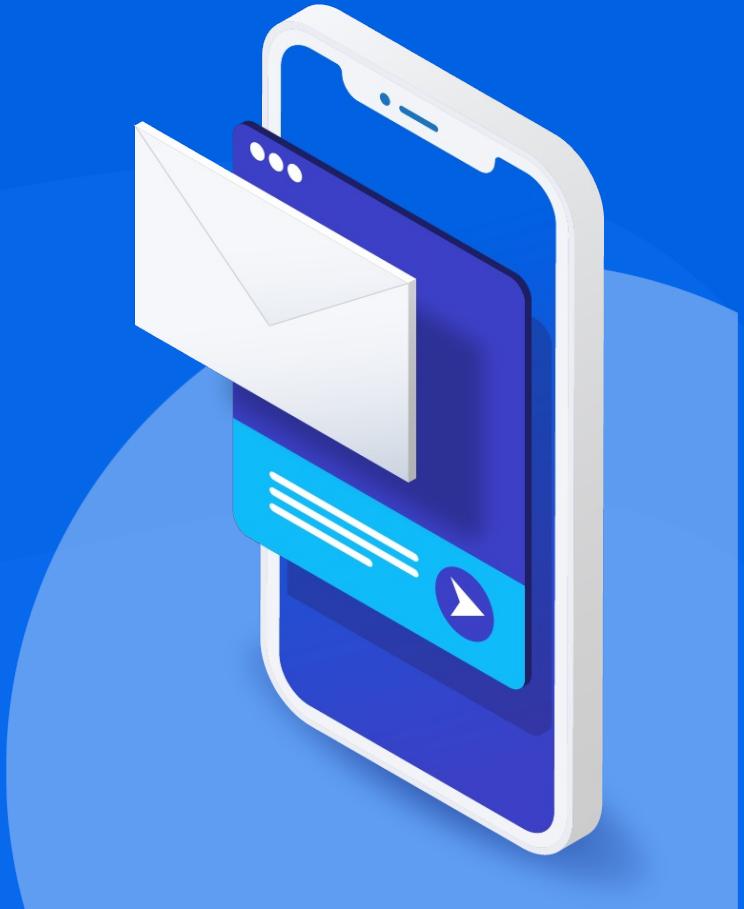


FEEDBACK

Any and all feedback is welcomed! We really do appreciate hearing from you!



support@cloudacademy.com



Thankyou!