



Amazon Alexa STEM workshop

2022 version

大綱

- Alexa 是什麼?
 - Alexa 在生活中可以扮演的角色
 - 它是真的人工智慧嗎?
- 註冊成為開發者與設定設備
- Alexa 是如何運作的?
 - 語意分析
- 建立自己的技能
- 挑戰: Plan the trip

Alexa 是什麼？



<https://www.youtube.com/watch?v=UOEIH2I9z7c>

Alexa 在生活中可以扮演的角色

■ Smart Home.

- Music / Podcast
- Phone / Story

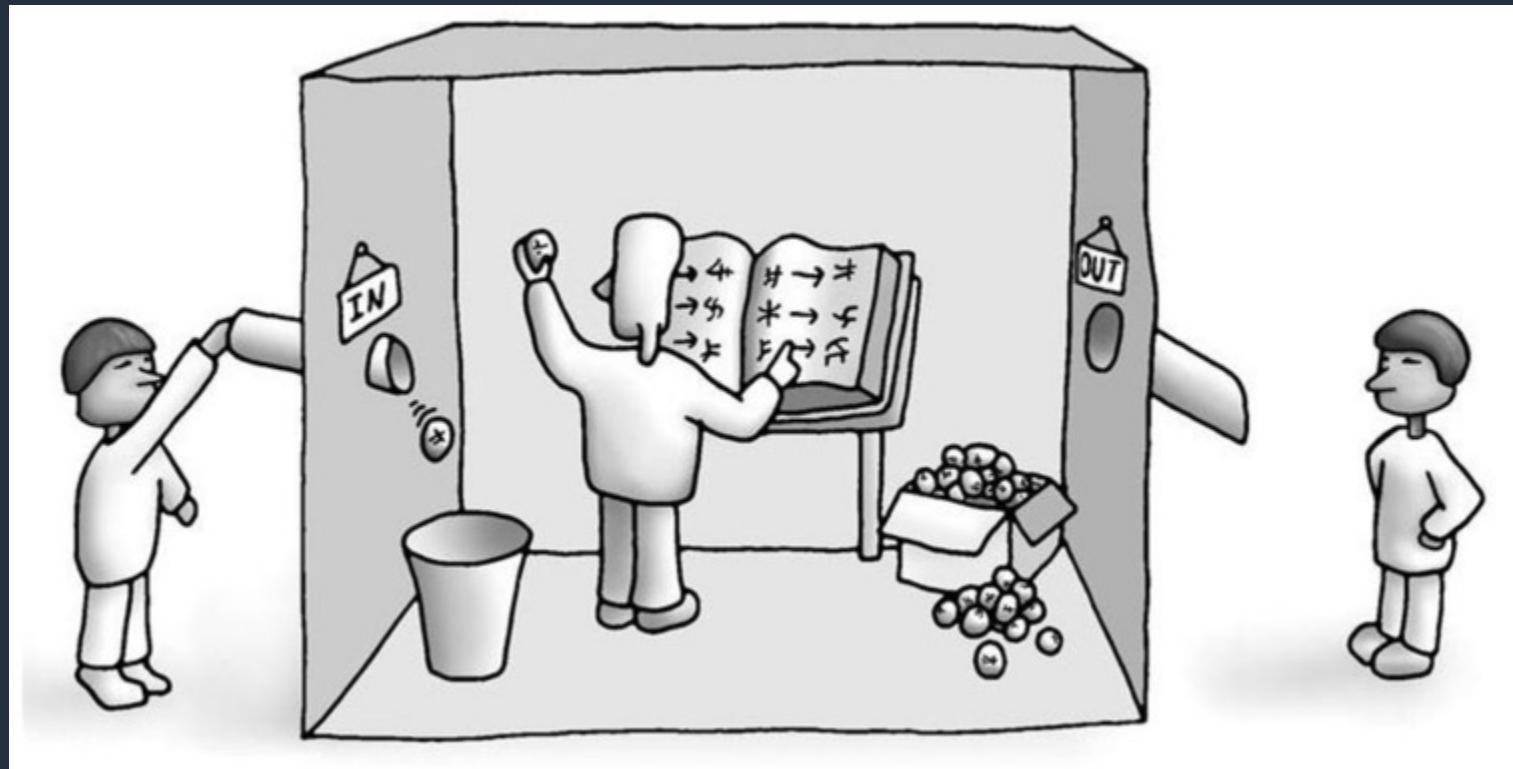


■ Business.

- Shopping !



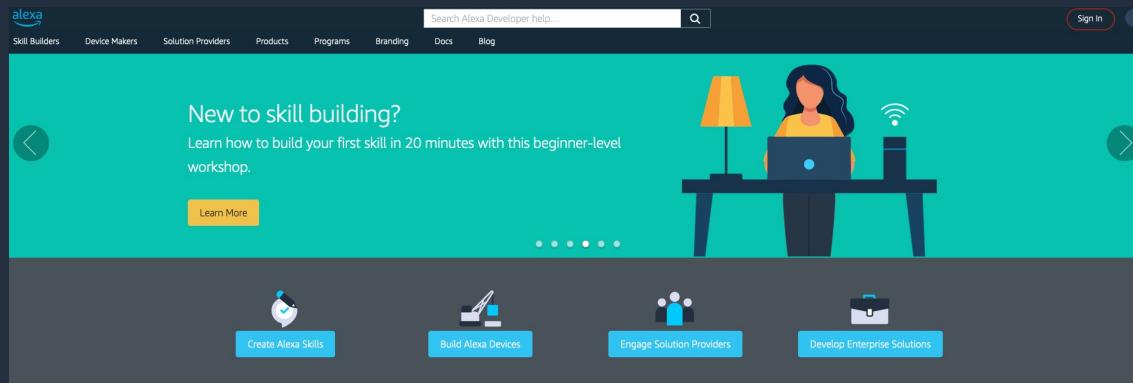
它真是人工智慧嗎？



讓我們開始設置吧！

註冊成為開發者 (1)

- 1) 打開 developer.amazon.com/alexa
- 2) 右上角點選 Sign In 按鈕
- 3) 點選 Create your Amazon account 的按鈕建立一個 Amazon 開發者帳號



Sign-In

Email or mobile phone number

Password [Forgot your password?](#)

Keep me signed in. [Details](#)

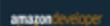
Sign-In

By continuing, you agree to Amazon's [Conditions of Use](#) and [Privacy Notice](#).

New to Amazon?

[Create your Amazon account](#)

註冊成為開發者 (2)



Amazon Developer Registration

Developer details

Sign up to...
• Publish your apps and skills across millions of devices in over 236 countries and territories.
• Build for natural voice experiences.
• Start generating revenue in a whole new way with Amazon's unique services and APIs.

First name*
TzuHsiang

Last name*
Lin

Country / Region*
Taiwan

Developer / Company name*
 * Developer / Company name is a required field.

Phone number*
 *

Email address*
Can be different from your login email, used as primary communication method.
*

Support email address

Same as above email address.

The name, email address and phone number entered above won't be displayed in Amazon appstore.

Contact Details

Developer / Company address*

Registration

1. Profile Information 2. Amazon Developer Services Agreement

English | 中文 (Chinese) | 日本語 (Japanese)

Last updated February 14, 2019

Current developers see what's changed.

Amazon Developer Services Agreement

This is an agreement between Amazon Digital Services LLC, Amazon Media EU S.r.l., Amazon Services International, Inc., Amazon Servicos de Varejo do Brasil Ltda., Amazon.com Int'l Sales, Inc., Amazon Australia Services, Inc., and Amazon Mexico Services, Inc. (each, individually, an "Amazon Party" and, together with their affiliates, "Amazon," "we" or "us") and you (if registering as an individual) or the entity you represent (if registering as a business) ("Developer" or "you"). Any other Amazon affiliate that we designate is also an Amazon Party.

- Structure of Agreement.** This agreement (the "Agreement") includes the body of the agreement below, all schedules to this agreement ("Schedules"), and all terms, rules and policies that we make available for participating in this program, including on our developer portal (together, the "Program Policies"). However, the terms in each Schedule only apply to you if you engage in the activity or use the Program Materials (defined in Section 3) to which the Schedule applies (for instance, the terms of the Distribution Schedule only apply to you if you submit a covered product to us to sell, distribute, or promote). Please carefully read the Agreement before clicking to accept it.
- Our Program.** Our program (the "Program") allows end users to purchase, download, and access software applications, games, and other digital products and services (for instance, the Amazon Alexa voice service (the "Alexa Service")) and allows developers to enable access to Amazon products and services in their Apps and Devices. "Apps" are software applications, games, and other digital products that you submit to us for sale, distribution, or promotion through the Program, or with which you use any Program Materials, together with their enhancements, upgrades, updates, bug fixes, new versions and other modifications and amendments. "Devices" are devices and device components that use any Program Materials. "Content" means your Apps, all content, ads, services, technology, data and other digital materials included in or made available through your Apps or Devices, and all

設定設備 (1)

1) 建立一個 Alexa 帳號

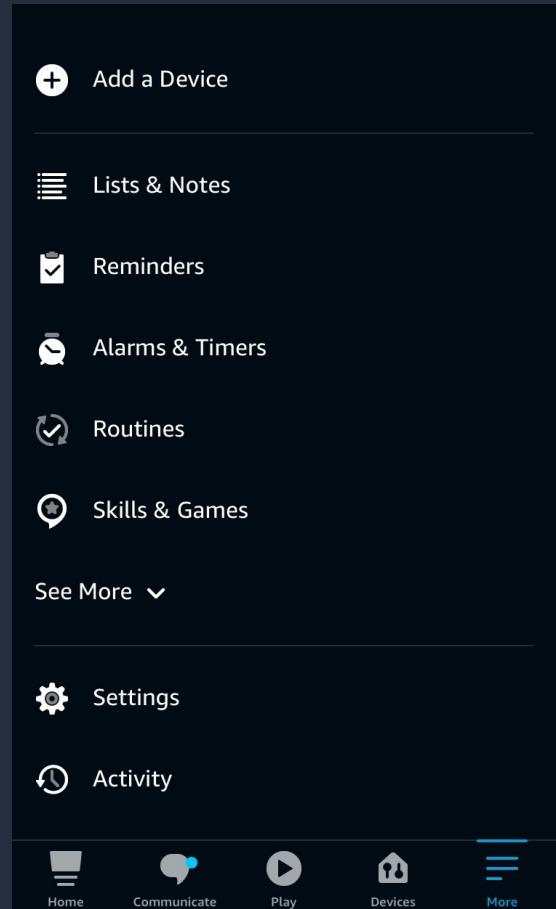
2) 從 Google Play 或 App Store 安裝 Alexa App

- <https://play.google.com/store/search?q=Alexa&c=apps&hl=zh-hk>
- <https://apps.apple.com/us/app/amazon-alexa/id944011620>

The screenshot shows the 'Create account' form for Alexa. It includes fields for 'First and last name', 'Your email address', and 'Create a password'. A note says 'Passwords must be at least 6 characters.' There is a 'Show password' checkbox and a large blue 'CREATE YOUR AMAZON ACCOUNT' button. Below the button, a note states: 'By creating an account, you agree to Amazon's [Conditions of Use](#) and [Privacy Notice](#)'. At the bottom, there is a link for 'Already have an account?' and a grey 'SIGN-IN NOW' button.

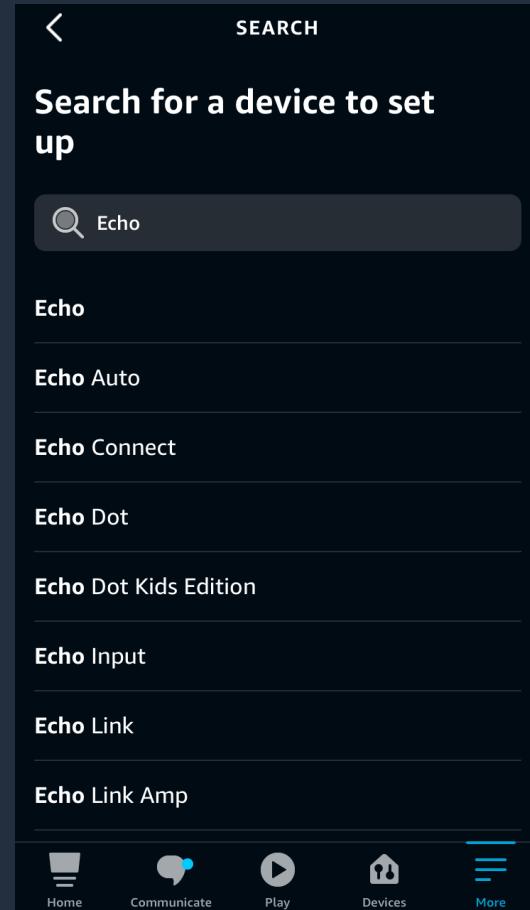
設定設備 (2)

- 1) 下方導覽列選擇 More
- 2) 點選 Add a Device



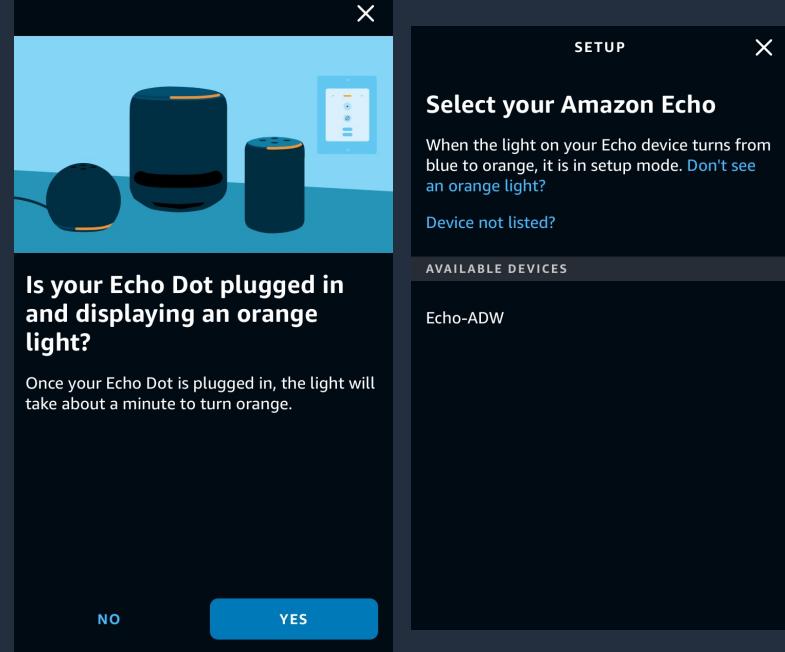
設定設備 (3)

- 1) 在搜尋的地方輸入 Echo
- 2) 點擊 Echo Dot

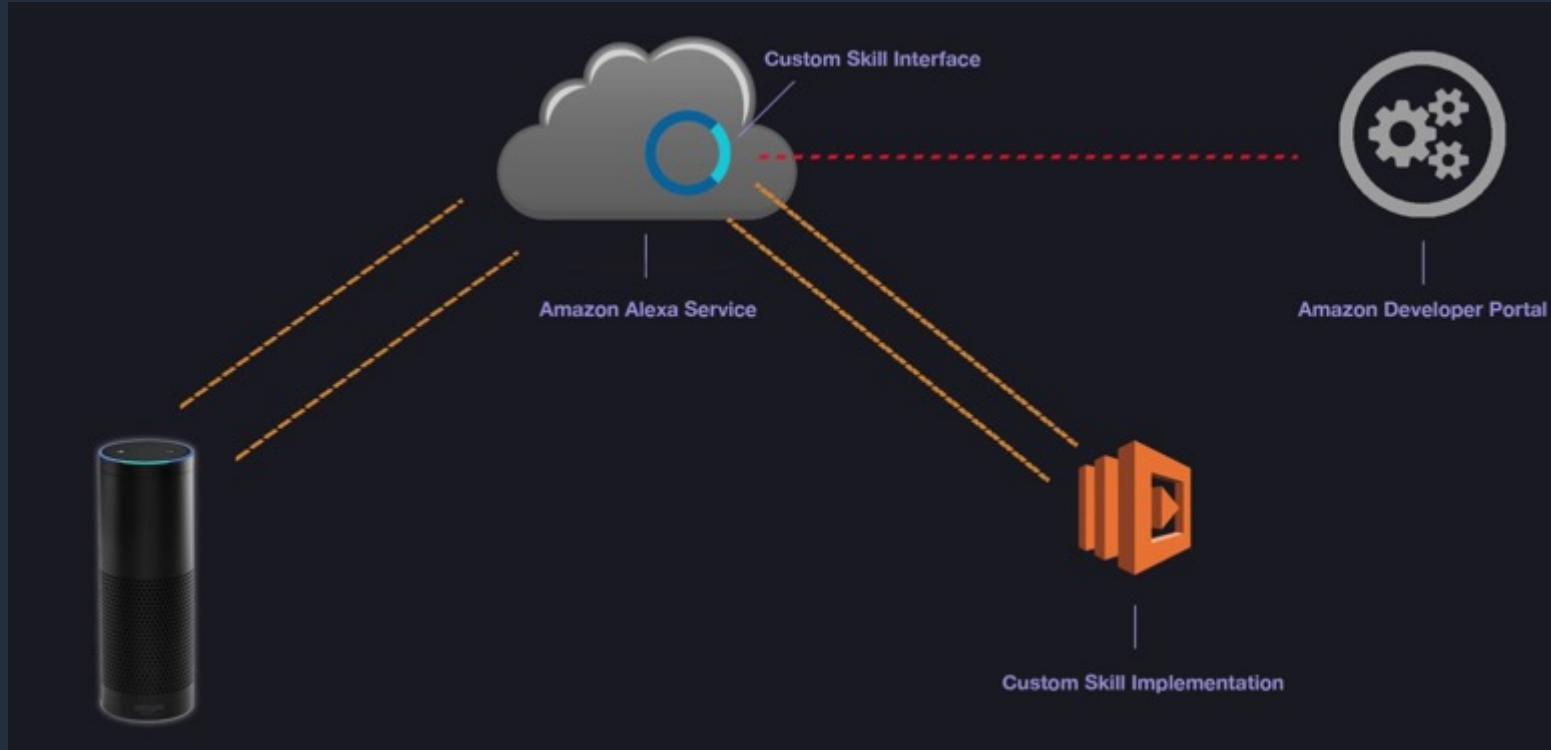


設定設備 (4)

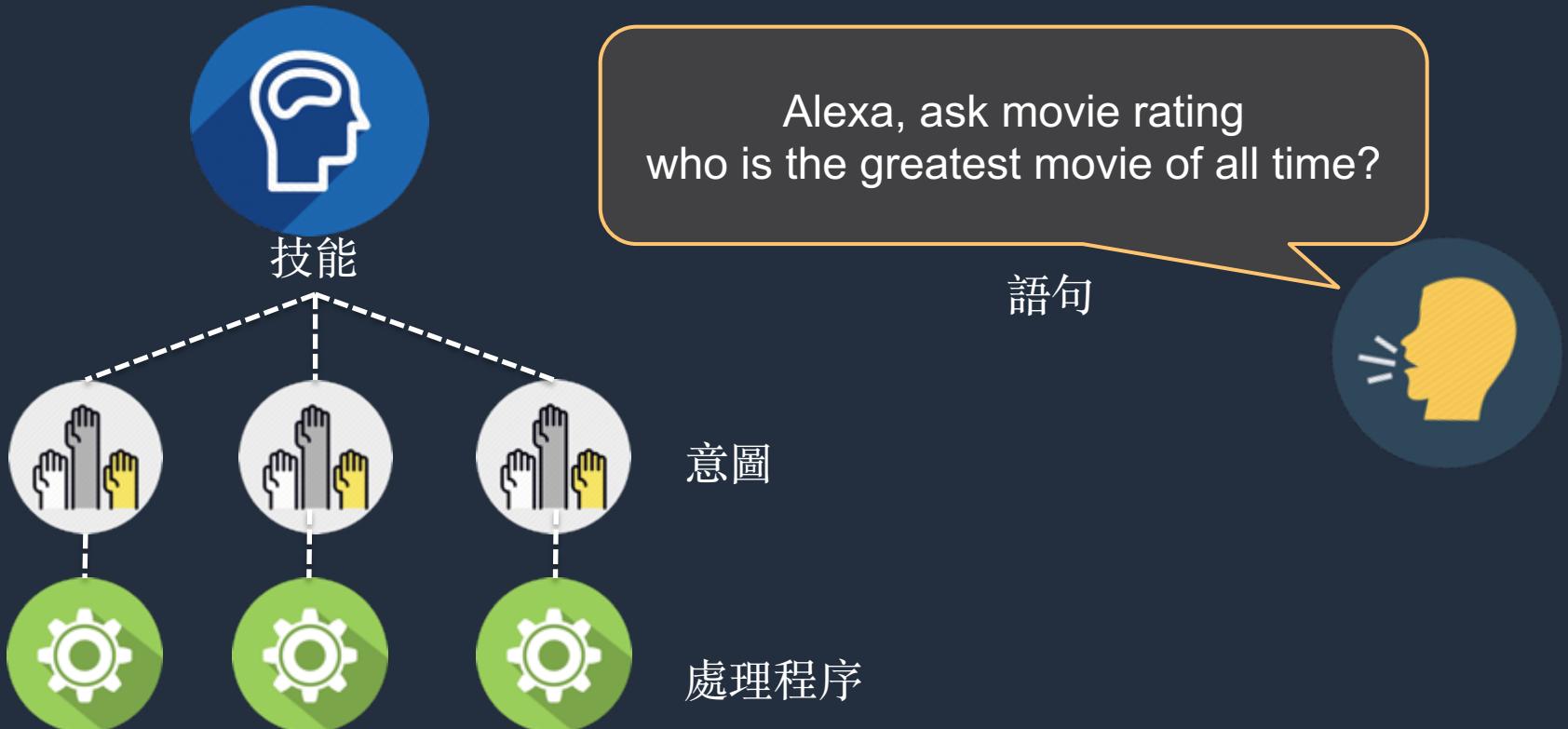
- 1) 接著會進入配對的地方
- 2) 我們要確保 Echo Dot 有橘燈環繞
- 3) 請按住 round Action (黑點) 大約 15 秒
- 4) 在 AVAILABLE DEVICES 選擇它



Alexa 是如何運作的?

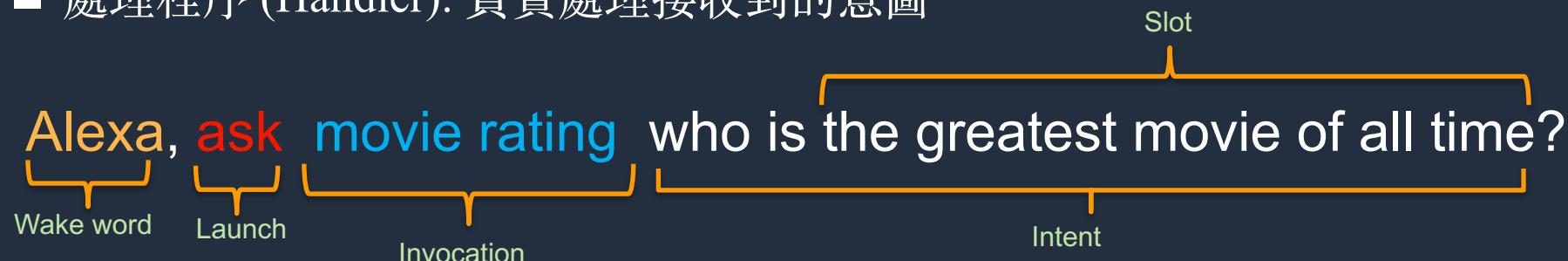


語意分析 (1)



語意分析 (2)

- 技能 (Skill): 就像 App Store 或 Google Play 裡面的一個 APP 一樣
- 喚醒 (Invocation): 與您的技能關聯的關鍵詞
- 發出的命令 (Launch): open, tell, launch 或 ask.
- 意圖 (Intent): 哪些動作滿足使用者說出的請求
- 語句 (Utterance): 哪些語句表現出意圖
- 插槽 (Slot): 在意圖中的參數
- 處理程序 (Handler): 負責處理接收到的意圖



語意分析 (3)

Alexa, ask movie rating who is the greatest movie of all time?

Wake word Launch Intent

Echo, tell plan my trip, I'd like to go to Japan.

Wake word Launch Invocation Intent Slot

Alexa, open plan my trip, I will go to Italy on September.

Wake word Launch Invocation Intent Slot Slot

讓我們開始建立自己的技能！

建立自訂技能 (1)

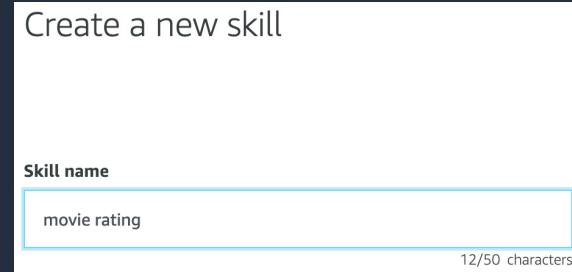
- 從 [developer.amazon.com/alex](https://developer.amazon.com/alexa) 開始
- 選 [Create Alexa Skills](#)



- 選右邊的 [Console](#)

建立自訂技能 (2)

- 1) 點擊 Create skill
- 2) 在 Skill name 的地方輸入 movie rating
- 3) 在選擇 model 的地方選擇 Custom 並且選擇 Alexa-hosted (Node.js)
- 4) 完成後，再選一次右上 Create skill



1. Choose a model to add to your skill

There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are intended and utterances that you can add to your skill.

Custom

Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.

Flash Briefing

Give users control of their news feed. This pre-built model lets users control what updates they listen to.

"Alexa, what's in the news?"

Smart Home

Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.

"Alexa, turn on the kitchen lights"

2. Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

Alexa-hosted (Node.js) **SELECTED**

Alexa will host skills in your account and get you started with a Node.js template. You will gain access to AWS Lambda endpoints in all Alexa service regions, a DynamoDB table for data persistence, and S3 for media storage. [Learn more](#)

Alexa-hosted (Python)

Alexa will host skills in your account and get you started with a Python template. You will gain access to AWS Lambda endpoints in all Alexa service regions, a DynamoDB table for data persistence, and S3 for media storage. [Learn more](#)

Provision your own

Provision your own endpoint and backend resources for your skill. This is recommended for skills that have significant data transfer requirements. You will not gain access to the console's code editor.

建立自訂技能 (3)

- 選擇 Start from Scratch
- 點擊 Continue with template 按鈕

Choose a template to add to your skill
Select a skill template from the list below or import a skill shared by the Alexa community as a public Git repository.

[Import skill](#) [Continue with template](#)

Start from Scratch <small>SELECTED</small> This skill gets you started with the required intents and with code demonstrating "Hello World" functionality if you are building an Alexa-hosted skill. Learn more By Alexa	Fact Skill Build an engaging fact skill about any topic. Alexa will select a fact at random and share it with the user when the skill is invoked. Learn more Includes: custom intents, Personalization By Alexa	High-Low Game Skill Try to guess a target number in a given range and Alexa will tell you if the number she had in mind was higher or lower. Learn more Includes: slots, custom intents, data persistence By Alexa	Pet Tales Skill Build a compelling multi-turn conversational audio and visual experience for a user looking for her favorite pet. Learn more Includes: API for Audio, API, custom intents, data persistence By Alexa	Fruit Shop Skill Build a multi-modal grocery shopping skill using custom and library controls for item lists, shopping cart management, and checkout. Learn more Includes: ASK SDK Controls Framework Preview, API, Personalization By Alexa
Scheduling Skill Build a skill to allow users to schedule appointments on your calendar, receive email confirmations and reminders. Learn more Includes: voice permissions, reminders, API calls, session persistence By Dabble Lab	Survey Skill Build a stand-up or survey skill that uses passcodes to allow only authorized users to provide updates and respond to questions. Learn more Includes: using passcodes, API calls, session persistence By Dabble Lab	Intro to Alexa Conversations This skill introduces you to Alexa Conversations by providing basic "Hello World" functionality and generating a voice response from Alexa. Learn more Includes: Alexa Conversations Preview, API, API for Audio, session persistence By Alexa	Weather Bot Skill Build a conversational weather bot skill that allows users to receive brief weather updates for a given location and date. Learn more Includes: Alexa Conversations, API for Audio, session persistence By Alexa	Pizza Ordering Example An example pizza ordering skill with Alexa Conversations demonstrating user corrections and context carryover. Learn more Includes: Alexa Conversations, API for Audio, session persistence By Alexa

Developer Console 簡介

- Build 用來設計意圖
- Code 用來設計意圖對應行為
- Test 用來執行和測試

The screenshot shows the Alexa developer console interface. At the top, there is a navigation bar with tabs: Your Skills, movie rating, Build, Code, Test, Distribution, Certification, and Analytics. The 'Test' tab is highlighted with a red box. Below the navigation bar is a search bar labeled 'Search Alexa Developer help...'. On the left side, there is a sidebar titled 'CUSTOM' with sections for Invocations, Interaction Model, Assets (which is expanded to show Slot Types (0)), Multimodal Responses, and Interfaces. The main content area has a title 'Alexa Design Guide' and text explaining the emergence of voice user interfaces (VUIs) like Amazon Alexa, noting it's a major advancement in human-computer interaction. It states that designing skills requires a different approach than traditional GUIs, focusing on conversational, voice-first interactions. A call-to-action at the bottom encourages visiting the Alexa Design Guide for tips and best practices. To the right, there is a 'Skill builder checklist' section with two items: 'Invocation Name' (marked as REQUIRED) and 'Intents, Samples, and Slots' (also marked as REQUIRED). Both items have green checkmarks indicating they are completed.

alexa developer console

Search Alexa Developer help...

Your Skills movie rating Build Code Test Distribution Certification Analytics

English (US)

CUSTOM

Invocations

Interaction Model

Assets

Slot Types (0)

Multimodal Responses

Interfaces

Alexa Design Guide

The emergence of voice user interfaces (VUIs), such as Amazon Alexa, isn't an incremental improvement to existing technology; it marks a major advancement in human-computer interaction. As such, designing skills differs from designing apps for traditional graphic user interfaces (GUIs). Rather than designing skills as replacements for GUIs, you need to change your whole design approach. Your skill designs must create truly conversational, voice-first interactions, while providing a harmonious GUI and touch-based interactions that are complementary to the main way users interact.

Visit the [Alexa Design Guide](#) for additional tips and best practices on how to design a satisfying and effective Alexa experience.

Skill builder checklist

REQUIRED

1. Invocation Name >

Enter an invocation name for your skill

REQUIRED

2. Intents, Samples, and Slots >

Add at least one intent and one sample utterance

建立自訂技能 (4)

1) 接著，我們要來定義 movie rating 的喚醒 (Invocation)

Invocations > Skill Invocation Name

Invocation

Users say a skill's invocation name to begin an interaction with a particular custom skill.
For example, if the invocation name is "daily horoscopes", users can say:

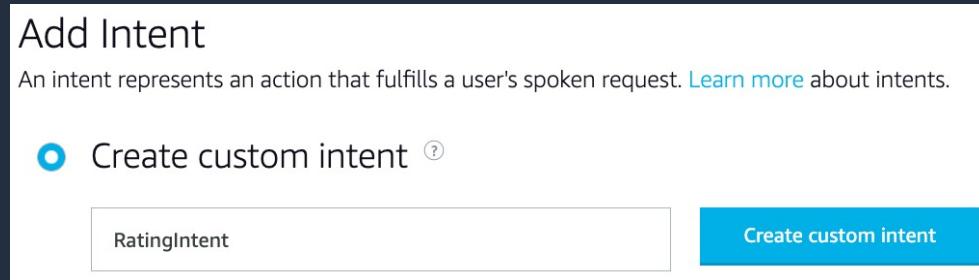
User: Alexa, ask daily horoscopes for the horoscope for Gemini

Skill Invocation Name ?
[How to pick names that are right for you](#)

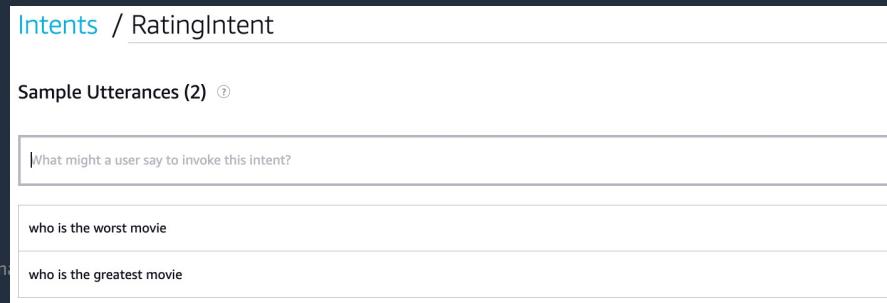
Brand names are only allowed if you provide proof of rights in the testing instructions or if you use the brand name in a referential manner that doesn't imply ownership (examples of terms that can be added to a brand name for referential usage: unofficial, unauthorized, fan, fandom, for, about).
If invocation name is an abbreviation, it should contain a space and a period after each letter (e.g.: a. b. c.).

建立自訂技能 (5)

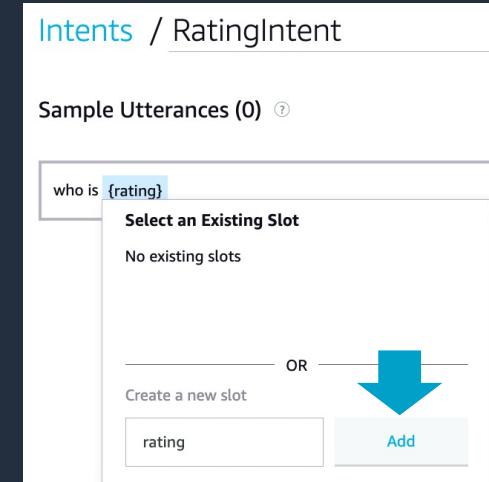
- 1) 增加自訂意圖來回答電影的排名 (movie rating), 像是 RatingIntent
Interaction Model > Intents > Add Intent



- 2) 輸入此意圖所有可能的語句 (Utterances)

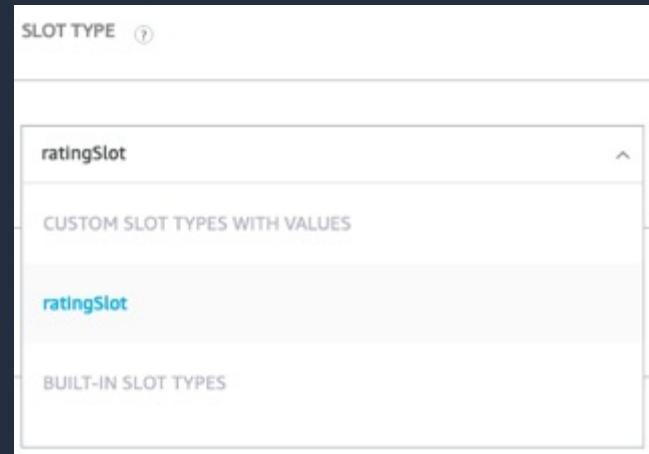
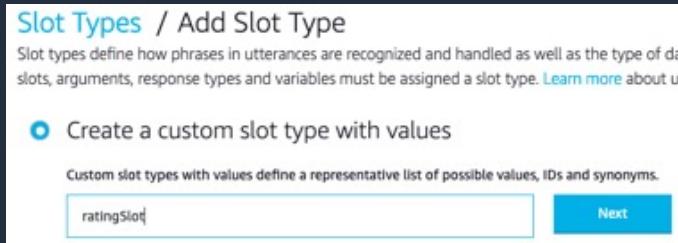


或定義 slot
簡化內容



建立自訂技能 (6)

- 1) 增加新的插槽類型 (Slot type), 像是 ratingSlot
- 3) 在 RatingIntent 中的插槽類型中加上它



- 2) 輸入這個插槽所有可能的值

VALUE	ID (OPTIONAL)
the worst movie of all time	Enter ID
the greatest movie of all time	Enter ID

利用 Amazon 已內建的意圖

Amazon Alexa 有一些已內建的意圖，可直接套用，例如：

Add Intent

An intent represents an action that fulfills a user's spoken request. [Learn more](#) about intents.

- Create custom intent ?

Create custom intent

- Use an existing intent from Alexa's built-in library ?

[Learn more](#) about using built-in intents.

🔍

1/147 built-ins

Name

Description



AMAZON.YesIntent

+ Add Intent



建立自訂技能 (7)

- 恭喜！已經完成意圖模型定義，按下 Save Model > Build model
- 接下來，根據請求類型 (request type) 和 意圖 (Intent) 來決定處理程式 (Handler)
 - LaunchRequest 發出命令 (Launch) 到 LaunchRequestHandler
 - IntentRequest 和意圖與 CancelIntent 或 StopIntent 相等，則會發出命令到 CancelAndStopIntentHandler
 - IntentRequest 和意圖為 RatingIntent，則發出命令到 RatingIntentHandler

選 Code 開始設計意圖對應程式

The screenshot shows the Alexa developer console interface. At the top, there's a navigation bar with tabs: 'Your Skills', 'movie rating', 'Build', 'Code' (which is highlighted with a red box), 'Test', 'Distribution', 'Certification', and 'Analytics'. To the right of the tabs is a search bar with the placeholder 'Search Alexa Developer help...' and a magnifying glass icon. Below the navigation bar is a toolbar with various icons: 'New File', 'New Folder', 'Delete', 'Rename', 'DynamoDB Database', 'S3 Storage', 'CloudWatch Logs', 'Usage', 'aws Integrate', 'Download Skill', 'Import Code', 'Offline Tools', and 'Docs'. On the left, there's a sidebar titled 'Skill Code' with a tree view showing files: 'Skill Code' (expanded), 'lambda' (expanded), 'index.js' (selected), 'local-debugger.js', 'package.json', and 'util.js'. The main content area displays the 'index.js' file's code:

```
1  /* *
2   * This sample demonstrates handling intents from an Alexa skill using the Alexa Skills Kit SDK (v2).
3   * Please visit https://alexa.design/cookbook for additional examples on implementing slots, dialog management,
4   * session persistence, api calls, and more.
5   */
6  const Alexa = require('ask-sdk-core');
7
8  const LaunchRequestHandler = {
9    canHandle(handlerInput) {
10      return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchRequest';
11    },
12    handle(handlerInput) {
13      const speakOutput = 'Welcome, you can say Hello or Help. Which would you like to try?';
14
15      return handlerInput.responseBuilder
16        .speak(speakOutput)
17        .reprompt(speakOutput)
18        .getResponse();
19    }
20  };
21
22  const HelloWorldIntentHandler = {
23    canHandle(handlerInput) {
24      return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
25        && Alexa.getIntentName(handlerInput.requestEnvelope) === 'HelloWorldIntent';
26    }
};
```

At the bottom right of the main content area, there's a small AWS logo.

建立自訂技能 (8)

- 1) 修改 LaunchRequestHandler
- 2) 當你說出發出命令的詞 (Launch word)，這表示你發出一個請求到 LaunchRequest。你將會進入到 LaunchRequestHandler 這個處理程序中。

```
const LaunchRequestHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchRequest';
  },
  handle(handlerInput) {
    const speakOutput = 'Welcome, you launch the movie rating intent. What can I help you?';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
};
```

```
return handlerInput.responseBuilder  
    .speak(speakOutput)  
.reprompt(speakOutput)  
    .getResponse();
```

呼叫 `reprompt` 函數是為了維持這一個對話 (`session`)

如果不呼叫 `reprompt` 則函數完成後就會結束對話

`reprompt` 令執行完函數後 Alexa 會繼續等待下一個輸入，這可以幫助我們輸入複雜的要求

建立自訂技能 (9)

- 1) 修改 CancelAndStopIntentHandler
- 2) 當你說出停止技能的詞 (stop skill word)，這表示你發出一個請求到 IntentRequest。如果意圖為暫停 (Stop) 或取消 (Cancel)，你將會進入 CancelAndStopIntentHandler 這個處理程序。

```
const CancelAndStopIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && (Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.CancelIntent'
        || Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.StopIntent');
  },
  handle(handlerInput) {
    const speakOutput = 'Leaving movie rating intent. Goodbye!';
    return handlerInput.responseBuilder
      .speak(speakOutput)
      .getResponse();
  }
};
```

建立自訂技能 (10)

- 為 RatingIntent 建立一個叫做 RatingIntentHandler 的處理程序
 - 複製整段 HelloWorldIntentHandler 開始

```
const RatingIntentHandler = {  
    canHandle(handlerInput) {  
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'  
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'RatingIntent'  
    },  
    handle(handlerInput) {  
        var rating = handlerInput.requestEnvelope.request.intent.slots.rating.value;  
        var speakOutput = 'Rating Intent !'  
  
        return handlerInput.responseBuilder  
            .speak(speakOutput)  
            .reprompt(speakOutput)  
            .getResponse();  
    }  
};
```

建立自訂技能 (11)

- 1) 在最下方 `export.handler = Alexa.SkillBuilders.custom().addRequestHandler` 的地方加上 `RatingIntentHandler`

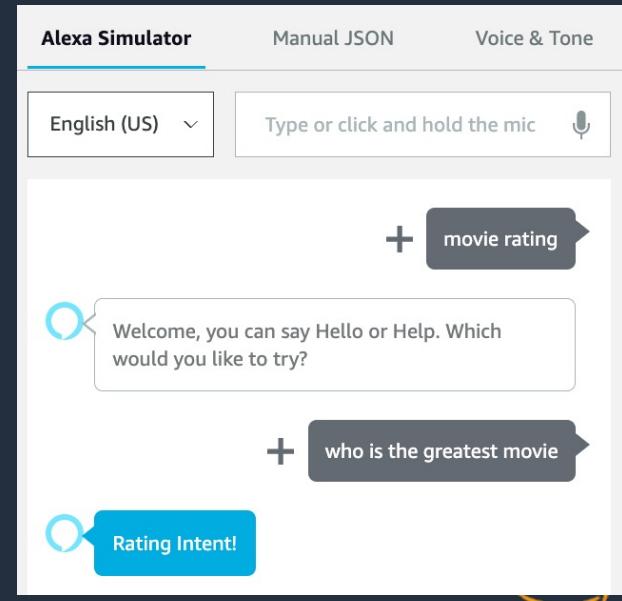
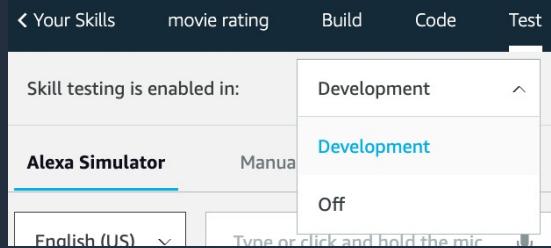
```
exports.handler = Alexa.SkillBuilders.custom()
  .addRequestHandlers(
    LaunchRequestHandler,
    HelloWorldIntentHandler,
    RatingIntentHandler, RatingIntentHandler,
    HelpIntentHandler,
    CancelAndStopIntentHandler,
    FallbackIntentHandler,
    SessionEndedRequestHandler,
    IntentReflectorHandler)
  .addErrorHandlers(
    ErrorHandler)
  .withCustomUserAgent('sample/hello-world/v1.2')
  .lambda();
```

- 2) 恭喜！完成程式設計，選擇 Save > Deploy

建立自訂技能 (12)

- 1) 每次你修改了你的意圖時，你都**必須**要點選 Build Model 來更新它
- 2) 每次你修改了你的處理程序時，你都**必須**要點選 Deploy 來部署它

- 接下來讓我們測試一下我們自訂的技能吧!!
 - 開啟測試功能



接下來讓我們把技能弄得更聰明!

我們要如何得知插槽的內容呢? (1)

- 我們先來查看 Skill I/O 的輸入內容

```
"request": {  
    "type": "IntentRequest",  
    "requestId": "amzn1.echo-api.request.f6984e7a-631c-4245-8663-c6c0f082e5c5",  
    "locale": "en-US",  
    "timestamp": "2022-05-06T05:55:25Z",  
    "intent": {  
        "name": "RatingsIntent",  
        "confirmationStatus": "NONE",  
        "slots": {  
            "rating": {  
                "name": "rating",  
                "value": "the greatest movie of all time",  
                "resolutions": {  
                    "resolutionsPerAuthority": [  
                        {  
                            "authority": "amzn1.er-authority.echo-sdk.amzn1.ask.skill",  
                            "status": {  
                                "code": "ER_SUCCESS_MATCH"  
                            },  
                            "values": [  
                                {  
                                    "value": {  
                                        "name": "the greatest movie of all time",  
                                        "id": "39c531a1d699320a719d7df3c5274147"  
                                    }  
                                }  
                            ]  
                        }  
                    ]  
                }  
            }  
        }  
    }  
}
```

我們要如何得知插槽的內容呢? (2)

- if...else

當條件成立的時候會執行 if 陳述式裡的程式，而不成立時則執行另外一個陳述式。

```
if(condition){  
    statement1  
} else {  
    statement2  
}
```

我們要如何得知插槽的內容呢? (3)

- If... else if...else

```
if(condition 1){  
    statement1  
} else if (condition 2) {  
    statement2  
} else {  
    statement3  
}
```

我們要如何得知插槽的內容呢? (4)

1) 我們來對 RatingIntentHandler 做一些修改!

```
const RatingIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'RatingIntent';
  },
  handle(handlerInput) {
    var rating = handlerInput.requestEnvelope.request.intent.slots.rating.value;
    var speakOutput = '';

    if(rating === 'the greatest movie'){
      speakOutput = `The greatest movie is Inception`;
    } else if(rating === 'the worst movie'){
      speakOutput = `The movie with the worst IMDb rating and number of IMDb votes of at least 73,292 is Reis at 1 out of 10.`;
    } else {
      speakOutput = `Sorry! I don't know what you're asking`;
    }

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
};
```

挑戰: Plan The Trip !



挑戰: Plan The Trip (1)

- 1) 讓 Alexa 幫你找到適合你的度假地點吧
- 2) Alexa 藉由與你的一問一答，找到可能適合你的度假地點
- 3) 你只需要回答 Yes 或 No，結束後最高分的地點就是最適合你的地點

"Do you like hot spring?",

"Do you like museum?",

"Is air quality important to you?",

"Do you want to have street food?",

"Do you want to go to America?",

"Do you want to have Asian food?",

"Do you want to visit top university?",

"Do you wish to encounter movie stars?",

"Do you want to explore languages other than Chinese and English?"

"Budapest",

"Mexico City",

"Hokkaido",

"Taipei",

"Heidelberg",

"Los Angeles"



挑戰: Plan The Trip (2)



挑戰: Plan The Trip (3-1)

<https://github.com/aws-ciel/alexa-plan-a-trip/blob/main/lambda/data.js>

問題

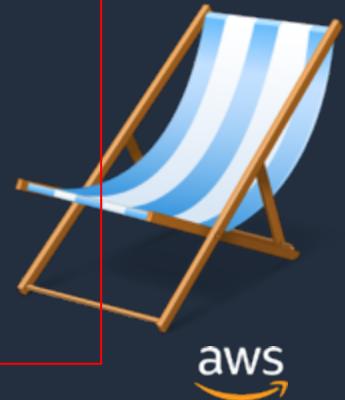
```
module.exports.questions = [
  "Do you like hot spring?",
  "Do you like museum?",
  "Is air quality important to you?",
  "Do you want to have street food?",
  "Do you want to go to America?",
  "Do you want to have Asian food?",
  "Do you want to visit top university?",
  "Do you wish to encounter movie stars?",
  "Do you want to explore languages other than Chinese and English?"
];
```

```
module.exports.destinations = [
  "Budapest",
  "Mexico City",
  "Hokkaido",
  "Taipei",
  "Heidelberg",
  "Los Angeles"
]
```

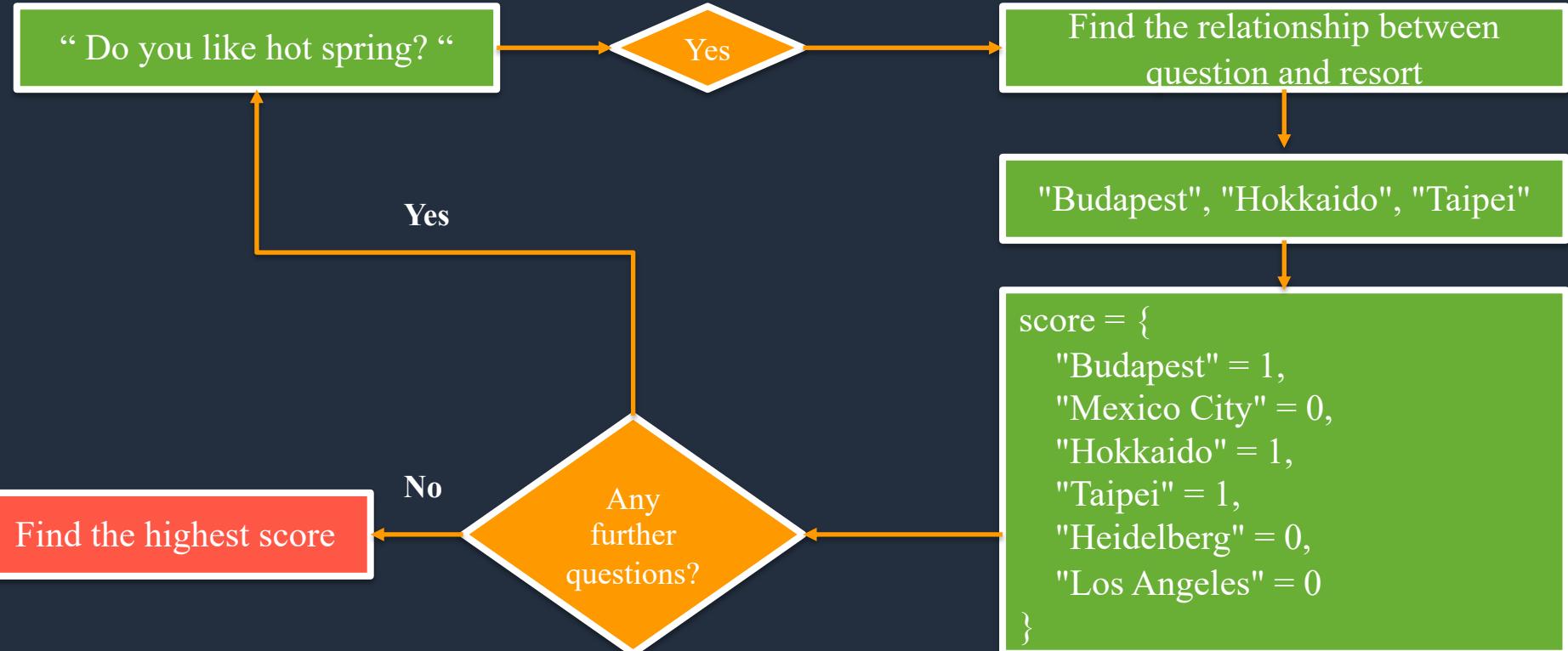
度假地點

問題與度假地點間的關聯性

```
module.exports.questionDestinationMatch = [
  ["Budapest", "Hokkaido", "Taipei"],
  ["Mexico City", "Los Angeles"],
  ["Budapest", "Hokkaido", "Heidelberg"],
  ["Mexico City", "Taipei"],
  ["Mexico City", "Los Angeles"],
  ["Mexico City", "Taipei", "Los Angeles"],
  ["Taipei", "Heidelberg", "Los Angeles"],
  ["Taipei", "Los Angeles"],
  ["Heidelberg", "Hokkaido"]
];
```



挑戰: Plan The Trip (3-2)



挑戰: Plan The Trip (4)

- 1) 點選 Create skill
- 2) 輸入 skill name 的地方輸入 plan the trip
- 3) 選擇 Custom model 和 Alexa-Hosted
- 4) 定義發出命令的關鍵字 (Launch keyword) 例 : summer trip
- 5) 建立一個名稱為 TripIntent 的意圖並且設置語句為 plan it 和 start planning
- 6) 建立一個預設意圖叫做 AMAZON.YesIntent 和 AMAZON.NoIntent。這兩個東西會幫忙處理 Yes 和 No 的回答。
- 7) 將問題和度假地點匯入你的 skill project 中

挑戰: Plan The Trip (5)

- 當你說出關鍵字 (summer trip) 時表示你送出一個請求到 LaunchRequest，你將會進入到 LaunchRequestHandler 這個處理程序。

```
const LaunchRequestHandler = {
  canHandle(handlerInput) {
    return handlerInput.requestEnvelope.request.type === 'LaunchRequest';
  },
  handle(handlerInput) {
    const speechText = 'Welcome to Plan the trip. I can find the best vacation destination for you. To start it, just say plan it or start planning.';
    return handlerInput.responseBuilder
      .speak(speechText)
      .reprompt(speechText)
      .getResponse();
  }
};
```

- 將問題和度假地點匯入到程式碼中:

```
let data = require('./data');
```

- 定義問題的數量:

```
const MAX_QUESTION_COUNT = data.questions.length;
```

挑戰: Plan The Trip (6)

1) for loop

```
for (initialExpression; conditionExpression; incrementExpression) {  
    // code block to be executed;  
}  
  
    for ( var i = 0; i < 10; i++) {  
        console.log( i );  
    }  
    for ( var i = 0; i < myArray.length; i++) {  
        console.log(myArray[i] );  
    }  
}
```

1) Array

```
var myArray = [“A”, “B”, “C”, “D”]
```

挑戰: Plan The Trip (7)

1) 字典 (Dictionary) 是一種用來儲存鍵值對 (key:value) 的一種資料結構

```
var scores = {  
    "Budapest": 0,  
    "Mexico City": 0,  
    "Hokkaido": 0,  
    "Taipei": 0,  
    "Heidelberg": 0,  
    "Los Angeles": 0  
}
```



```
console.log( scores["Hokkaido"] );  
console.log( scores["Los Angeles"] );
```

挑戰: Plan The Trip (8)

<https://github.com/aws-ciel/alexa-plan-a-trip/blob/main/lambda/index.js>

- 1) 為 TripIntent 建立一個處理程序 (Handler)
- 2) 當你說出 plan it 或 start planning 表示你發出一個請求到 IntentRequest。如果意圖為 TripIntent 就會進入 TripIntentHandler 的處理程序。

```
const TripIntentHandler = {
    canHandle(handlerInput) {
        return handlerInput.requestEnvelope.request.type === 'IntentRequest'
            && handlerInput.requestEnvelope.request.intent.name === 'TripIntent';
    },
    handle(handlerInput) {
        // Initial the score for all resorts.
        // scores = {"Budapest":0, "Mexico City":0, "Hokkaido": 0, "Taipei": 0, "Heidelberg": 0, "Los Angeles":0 }
        let scores = {};
        for(var i=0; i<data.destinations.length; i++){
            var location = data.destinations[i];
            scores[location] = 0
        }

        // Initial the content
        let attrs = handlerInput.attributesManager.getSessionAttributes();
        attrs.questionCount = 0; // The first question
        attrs.scores = scores; // The score of all resorts currently

        // Start with the first question
        let question = data.questions[0];
        // return the first question
        var speechText = `Ok. Let's start it. ${question}`;
        return handlerInput.responseBuilder
            .speak(speechText)
            .reprompt(question)
            .getResponse();
    }
};
```

挑戰: Plan The Trip (9-1)

- 1) 為 YesIntent 和 NoIntent 建立一個叫作 YesNoIntentHandler 的處理程序。
- 2) 當你說出 Yes 或 No 表示你發出請求到 IntentRequest。如果意圖為 YesIntent 或 NoIntent 則會進入到 YesNoIntentHandler 這個處理程序中。

```
const YesNoIntentHandler = {  
    canHandle(handlerInput) {  
  
        //If your answer is Yes or No  
        return handlerInput.requestEnvelope.request.type === 'IntentRequest'  
            && ['AMAZON.YesIntent', 'AMAZON.NoIntent'].includes( handlerInput.requestEnvelope.request.intent.name);  
    },  
    handle(handlerInput) {  
        const responseOutput = handlerInput.responseBuilder.  
            speak('Yes or No')  
            .getResponse();  
        return responseOutput;  
    }  
};
```

挑戰: Plan The Trip (9-2)

```
const YesNoIntentHandler = {
    canHandle(handlerInput) {

        //If your answer is Yes or No
        return handlerInput.requestEnvelope.request.type === 'IntentRequest'
            && ['AMAZON.YesIntent', 'AMAZON.NoIntent'].includes( handlerInput.requestEnvelope.request.intent.name);
    },
    handle(handlerInput) {

        // Fetch the content of our last conversation with Alexa
        let attrs = handlerInput.attributesManager.getSessionAttributes();
        let currentQuestionCount = attrs.questionCount;
        let scores = attrs.scores; // The score of all resorts currently

        // Check the answer. If Yes, +1 score for corresponding resort
        if(handlerInput.requestEnvelope.request.intent.name == 'AMAZON.YesIntent'){
            // Find the corresponding resort
            let destinationMatch = data.questionDestinationMatch[currentQuestionCount];
            console.log(destinationMatch);
            // +1 score for corresponding resort(s)
            for(var i=0; i<destinationMatch.length; i++){
                var location = destinationMatch[i];
                scores[location] = scores[location] + 1;
            }
        }

        // To the next question and update the score
        attrs.questionCount = currentQuestionCount + 1;
        attrs.scores = scores;
    }
}
```

挑戰: Plan The Trip (9-3)

```
// Check if available questions exist
if (attrs.questionCount >= MAX_QUESTION_COUNT) {
    // Find the highest score resort
    var topDestination = '';
    var topScore = -1;
    for(var j=0; j<data.destinations.length; j++){
        var destination = data.destinations[j];
        let score = scores[destination];
        if (score > topScore) {
            topDestination = destination;
            topScore = score;
        }
    }
    // return the result
    let speechText = `Based on my calculation. You'll enjoy ${topDestination} for your next vacation. Thanks for playing Plan the trip.`;
    return handlerInput.responseBuilder
        .speak(speechText)
        .getResponse();
}

} else {
    // continue to ask the next question
    let question = data.questions[currentQuestionCount + 1];
    let speechText = `Next question. ${question}`;
    return handlerInput.responseBuilder
        .speak(speechText)
        .reprompt(speechText)
        .getResponse();
}
};

};
```

挑戰: Plan The Trip (10)

- 1) 在最下方 `export.handler = Alexa.SkillBuilders.custom().addRequestHandler` 的地方加上 `TripIntentHandler` 和 `YesNoIntentHandler`

```
exports.handler = Alexa.SkillBuilders.custom()
  .addRequestHandlers(
    TripIntentHandler,
    YesNoIntentHandler,
    LaunchRequestHandler,
    HelpIntentHandler,
    CancelAndStopIntentHandler,
    SessionEndedRequestHandler,
    IntentReflectorHandler) // make sure IntentReflectorHandler is last so it doesn't override your custom intent handlers
  .addErrorHandlers(
    ErrorHandler)
  .lambda();
```

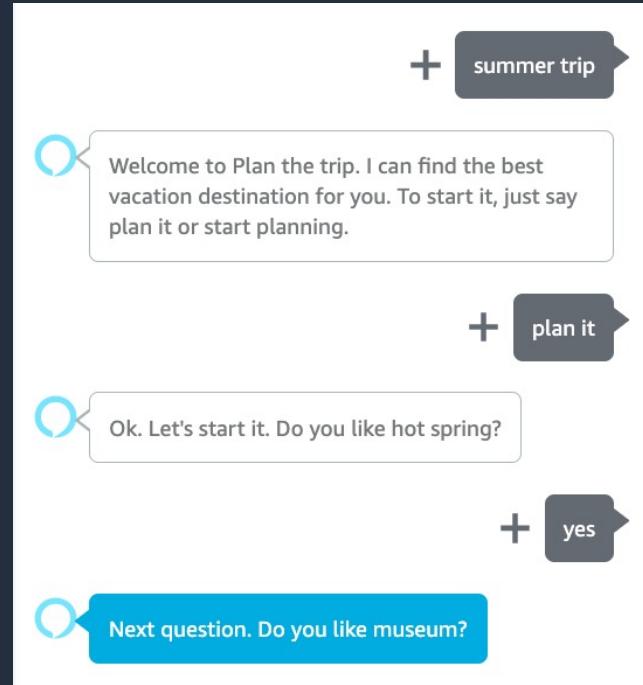
挑戰: Plan The Trip (11)

1) 測試時間!!

- Test ---> Development

2) 輸入 ask plan the trip ---> plan it

3) 我們可以查看 JSON 輸出來除錯



我們需要你們的回饋!