

Getting an Amazon developer account

1. Go to <https://goo.gl/WBs8xf>
2. Enter your name and email (can be school or personal email, but it is better to use your school one) and choose a password (remember it!). Then click the yellow 'Create your Amazon Developer account' button
3. Next, fill in the fields with a red asterisk beside them:

Email: Your school or personal one, but better to use school one

Phone number: Your own one or 07700 900000

Developer name or company name: 'Freelancer' or your own name

Address: Put your home address or the school's address. Better to use your home one as that's where they'll send any gifts e.g. t-shirts you might win

If you still want to use the school address, it's:

Address 1: City of London School

Address 2: 107 Queen Victoria St

City: London

State/Province/Region: London

Zip code/Postal code: EC4V 3AL

Once done, click the yellow 'Save and continue' button at the bottom of the page:

1. Profile Information

2. App Distribution Agreement

3. Payments

* Indicates a required field.

Country/Region *	United Kingdom
First name *	Vinu
Last name *	Jey
Email address *	v.jeyananthan@cityoflondonschool.org.uk
Phone number *	07522924649
e.g. 212-555-1212, +44 0161 715 3369	
Fax number	
Developer name or company name *	Freelancer
Displayed on your apps at Amazon.com	Required field
Developer description	
Maximum characters: 4000, Remaining: 4000	
Address 1 *	136, The Heights
Address 2	
City *	London
State/Province/Region *	Central London
Zip code/Postal code *	UB5 4BT
Customer support email address	
Customer support phone	
Customer support website	

Cancel

Save and Continue

4. Read through the "App Distribution Agreement", if you agree scroll to the bottom and hit the "Accept and Continue" button

5. Finally, choose no for both options on the “Payments” tab, then click on “Save and Continue”:

Registration

✓ 1. Profile Information

✓ 2. App Distribution Agreement

3. Payments

* Indicates a required field.

Do you plan to monetize your digital content, such as charging for apps or games or selling in-app items or in-game items, or by receiving cash rewards for your skills? *

No

Yes

Do you plan to monetize apps by displaying ads from the Amazon Mobile Ad Network or Mobile Associates? *

No

Yes

Note: You may still monetize later if you select "No" by entering payment and tax information from the Settings menu.

Cancel

Save and Continue

Creating an AWS Educate Starter Account

1. Go to <https://goo.gl/XCnSnB>
2. Sign up with:

Institution name: City of London School

Country: United Kingdom

City: London

State: (leave blank)

Field of Study: Other

Email: Your school email. This is your first initial DOT your last name @cityoflondonschool.org.uk

Grade Level: Vocational/Community college

Grad Month: 08

Grad Year: 2025 - your year

OG: 2025, 1st form: 2024, 2nd form: 2023, 3rd form: 2022, 4th form: 2021...

Promo Code: (leave blank)

3. Select “AWS Educate Starter Account”, then click “Next”
4. Check your school email to verify your account.



Apply to join AWS Educate

Step 2/3: Tell us about yourself

City of London School

Start typing the name of your school and select from the list. If you don't see your school, enter the full name, example: Harvard University

United Kingdom

London

State

Adam

Jones

Other

a.jones@cityoflondonschool.org.uk

Please provide a valid, current email issued by your institution. Example: your_name@your_school.edu

Vocational/Community College

08

2018

Promo Code

[Frequently Asked Questions](#)



Apply to join AWS Educate

Step 3/3: Choose one of the following

☐ Click here to enter an AWS Account ID

Approved students are sent a welcome email and benefits including and AWS promotional code.

Don't have one? Sign up now

☒ Click here to select an AWS Educate Starter Account

An AWS Educate Starter Account is a free, capped-account that doesn't require a credit card. There are some Usage limitations including an approximately 25% reduction in access to AWS services. Because Starter Accounts are capped, a separate AWS promotional code is not provided.

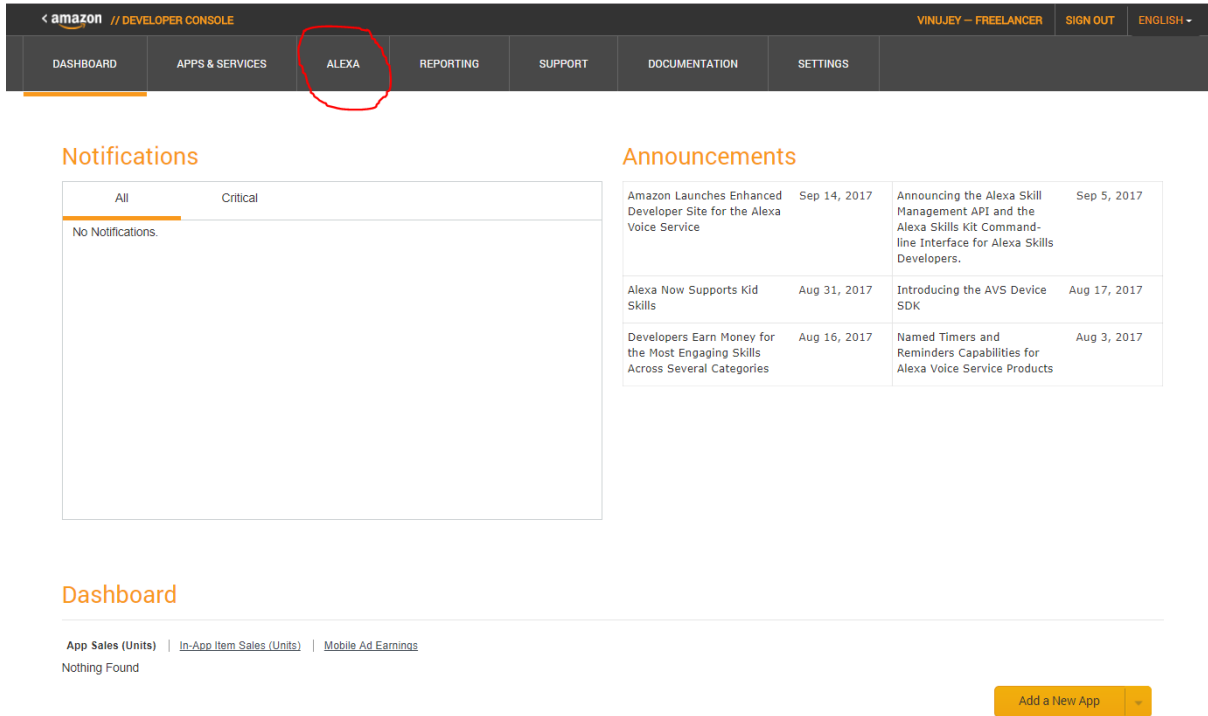
[Frequently Asked Questions](#)

With a Starter Account you will receive access to AWS Educate and a lab account with your usage of AWS services capped at the lab amount. A separate AWS promotional credit will not be provided to you.

NEXT

Starting an Alexa Skills Kit skill

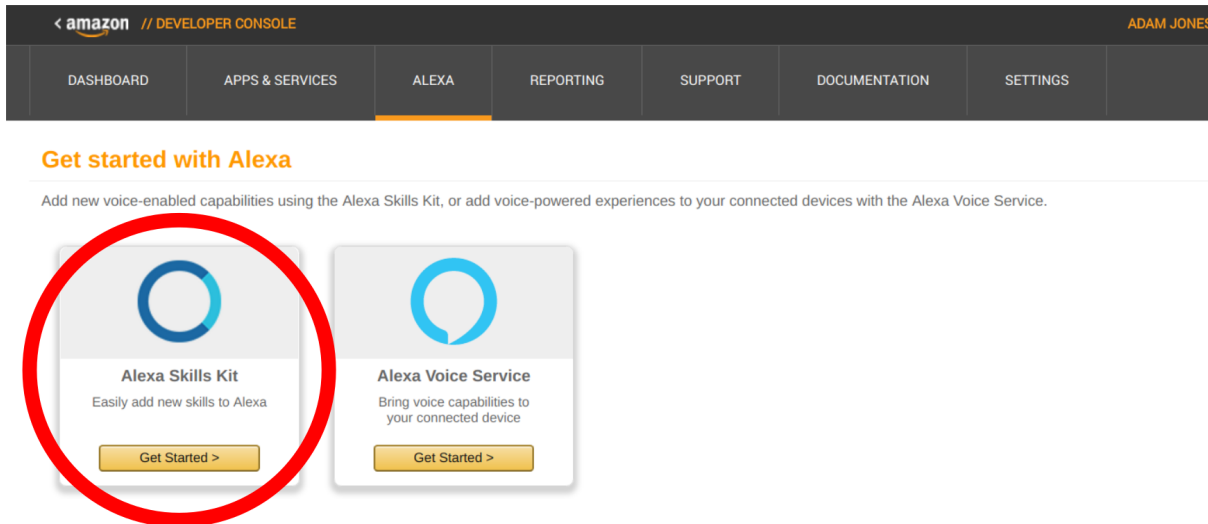
1. Go to <https://goo.gl/DwfySL>
2. Sign in with your Amazon Developer account created previously if necessary
3. You will be presented with this screen, on the navbar at the top, select “Alexa”:



The screenshot shows the Amazon Developer Console interface. The top navigation bar includes links for DASHBOARD, APPS & SERVICES, ALEXA (highlighted with a red circle), REPORTING, SUPPORT, DOCUMENTATION, and SETTINGS. Below the navigation bar, there are sections for Notifications and Announcements. The Notifications section shows 'No Notifications.' The Announcements section displays a table of recent updates. At the bottom, there is a 'Dashboard' section with links for App Sales (Units), In-App Item Sales (Units), and Mobile Ad Earnings, and a button to 'Add a New App'.

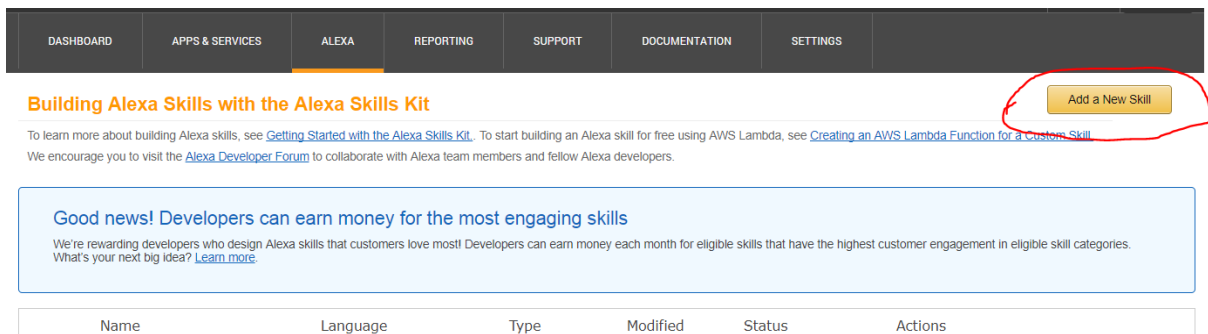
Announcement	Date	Announcement	Date
Amazon Launches Enhanced Developer Site for the Alexa Voice Service	Sep 14, 2017	Announcing the Alexa Skill Management API and the Alexa Skills Kit Command-line Interface for Alexa Skills Developers.	Sep 5, 2017
Alexa Now Supports Kid Skills	Aug 31, 2017	Introducing the AVS Device SDK	Aug 17, 2017
Developers Earn Money for the Most Engaging Skills Across Several Categories	Aug 16, 2017	Named Timers and Reminders Capabilities for Alexa Voice Service Products	Aug 3, 2017

4. Next click “Get Started” underneath the “Alexa Skills Kit” header



The screenshot shows the 'Get started with Alexa' section of the Amazon Developer Console. It features two cards: 'Alexa Skills Kit' and 'Alexa Voice Service'. The 'Alexa Skills Kit' card is highlighted with a red circle and contains the text 'Easily add new skills to Alexa' and a 'Get Started >' button. The 'Alexa Voice Service' card contains the text 'Bring voice capabilities to your connected device' and a 'Get Started >' button.

5. Click on “Add a new skill” in the top right



The screenshot shows the 'Building Alexa Skills with the Alexa Skills Kit' section of the Amazon Developer Console. It includes a header with the title 'Building Alexa Skills with the Alexa Skills Kit' and a button to 'Add a New Skill' (highlighted with a red circle). Below the header, there is a paragraph of text and a table with columns for Name, Language, Type, Modified, Status, and Actions.

Good news! Developers can earn money for the most engaging skills

We're rewarding developers who design Alexa skills that customers love most! Developers can earn money each month for eligible skills that have the highest customer engagement in eligible skill categories. What's your next big idea? [Learn more](#)

Name	Language	Type	Modified	Status	Actions
------	----------	------	----------	--------	---------

6. You should now be at this screen:

Skill Type: Custom Interaction Model

Language: English (U.K.)

Name: Choose the name for your skill

Invocation Name: Choose what you say for Alexa to start your skill - something unique and easily pronounceable and ideally use common words (check with us at this point - just put your hand up)

Audio Player: No

Video App: No

Render Template: No

Once you've filled in the fields, hit **Save** at the bottom, then hit the yellow **Next** button

DASHBOARD

APPS & SERVICES

ALEXA

REPORTING

SUPPORT

DOCUMENTATION

SETTINGS

< Back to All Skills

Create a New Alexa Skill

Skill Information

Interaction Model

Configuration

SSL Certificate

Test

Publishing Information

Privacy & Compliance

Skill Type
Define a custom interaction model or use one of the predefined skill APIs. [Learn more](#)

☒ Custom Interaction Model

☐ Smart Home Skill API

☐ Flash Briefing Skill API

☐ Video Skill API

Language
Language of your skill

English (U.K.) ▾

Name
Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.

Invocation Name
The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...".

For successful Alexa Skills Certification, please review and follow our [Invocation Name Guidelines](#) as well as our [Certification Requirements](#).

Global Fields
These fields apply to all languages supported by the skill.

Audio Player
Does this skill use the audio player directives? [Learn more](#)

☐ Yes

☒ No

Video App
Does this skill use the video app directives? [Learn more](#)

☐ Yes

☒ No

Render Template
Does this skill use the Render Template directives? [Learn more](#)

☐ Yes

☒ No

Save

7. You will then be met with this screen, where you will click on **Launch Skill Builder**

The screenshot shows the Amazon Skills Developer console for a skill named "Weather". The top navigation bar includes links for DASHBOARD, APPS & SERVICES, ALEXA (highlighted), REPORTING, SUPPORT, DOCUMENTATION, and SETTINGS. Below the navigation bar, there's a "Back to All Skills" link and a skill card for "Weather" with a custom ID. A sidebar on the left lists various configuration sections: Skill Information, Interaction Model, Configuration, Test, Publishing Information, and Privacy & Compliance, each with a status icon. A "Skills Beta Testing" section indicates the skill is not yet eligible. The main content area features a banner for the "Launch Skill Builder (beta)" interface, which is circled in red. Below the banner, there's an "Intent Schema" section with a text area for defining intents. Further down, there's a "Custom Slot Types" section with an "Enter Type" input field and an "Enter Values" text area for defining slot values.

8. Once you are on the below screen, click on **Add an intent**

The screenshot shows the Amazon Skills Developer console dashboard. The left sidebar contains links for Dashboard, Code Editor, Intents (3), and Slot Types (0). The main content area is titled "Dashboard" and features two primary cards: "Intents" and "Slot Types". The "Intents" card shows 3 built-in intents and 0 custom intents, with an "Add an Intent" button circled in red. The "Slot Types" card shows 0 built-in slot types and 0 custom slot types, with an "Add a Slot Type" button. Below these cards, there's a section for "Synonyms and IDs for Slot Type Values (Entity Resolution)" with a table listing values like "bacon", "mushrooms", "olives", and "pineapple" along with their IDs and synonyms. A "Read more" link is provided. At the bottom, there's a "How to get started" section with three video thumbnails: "Part 1: An introduction to dialog interfaces", "Part 2: Using intents, built-ins and utterances", and "Part 3: Simplifying code for multi-turn dialogs".

9. Create two new **custom intents** by entering the names you want for them on this page:

The 'Add intent' dialog shows two options: 'Create a new custom intent' (selected) and 'Use an existing intent from the built-in library'. The 'Create a new custom intent' option has a text input field with the placeholder 'Enter the name of the new intent...' and a 'Create intent' button. The 'Use an existing intent from the built-in library' option has a search bar labeled 'Search built-in intents'. Below the search bar, a table lists built-in intents. The first row is 'Standard' with 22 built-in intents, described as 'The standard intents are used for common, general actions such as stopping, canceling, and asking for help.' The table has columns for 'NAME' and 'DESCRIPTION'.

They can be anything you like but we suggest making one that is about the user being **positive** about something and one that is the user being **negative** about something.

The 'Positive' intent configuration page shows a search bar for 'Sample Utterances (7)' and a list of sample utterances: 'good', 'best', 'favorite club', 'programming club is fun', 'fun', 'love', and 'adore'. Each utterance has a trash icon to its right. The 'Negative' intent configuration page shows a search bar for 'Sample Utterances (8)' and a list of sample utterances: 'bad', 'not enough donuts', 'no donuts', 'waste of time', 'boring', 'I hate programming club', 'dislike', and 'hate'. Each utterance has a trash icon to its right.

Intents are ideas that your program will match. For example, here we have two intents; called Positive and Negative. The **positive** one will match someone saying something **positive** about programming club, and the **negative** one will match someone saying something **negative** about programming club.

We specify **sample utterances** as examples of what someone might say to trigger the intent, for example 'programming club is fun' and 'programming is my favourite club' are sample utterances for **positive**, and 'programming is boring' and 'not enough donuts' are sample utterances for **negative**.

Finally, hit **Save Model** and **Build Model** at the top:

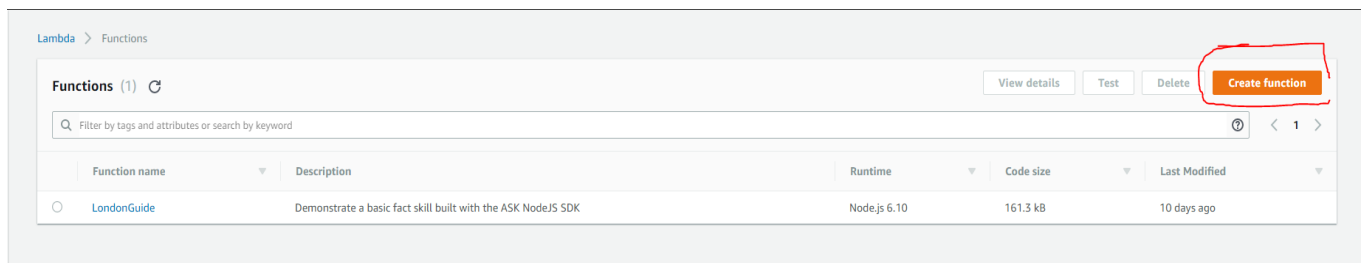
The bottom of the chatbot interface shows a dark bar with the text 'Weather' and 'English (UK)' on the left. On the right, there are two buttons: 'Save Model' (with a floppy disk icon) and 'Build Model' (with a wrench and hammer icon). These two buttons are circled in red.

Writing your code:

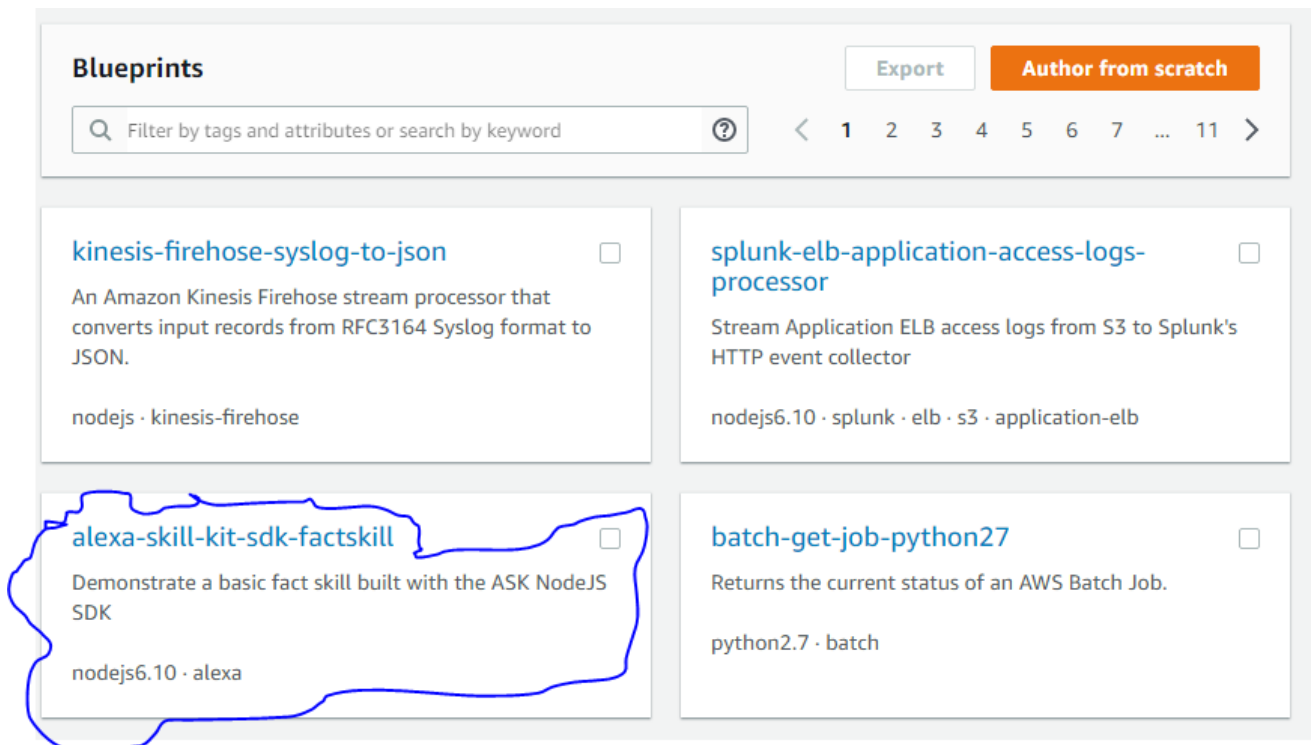
1. Select **all** the code on this page and copy it:
<https://goo.gl/6BCCFs>
2. Paste it into a text editor. If you're not sure what to use you can try <http://requirebin.com/>
3. Have a look at the code. The lines beginning // are comments, which means the computer doesn't read them but they're there to help you understand what the code does. If you don't know JavaScript try deciphering what the code does. There's a good JavaScript quick tutorial at <http://jsforcats.com/> (ignore the stuff about 3rd party libraries). Try editing the code so it'll say what you want it to say.

Running your code:

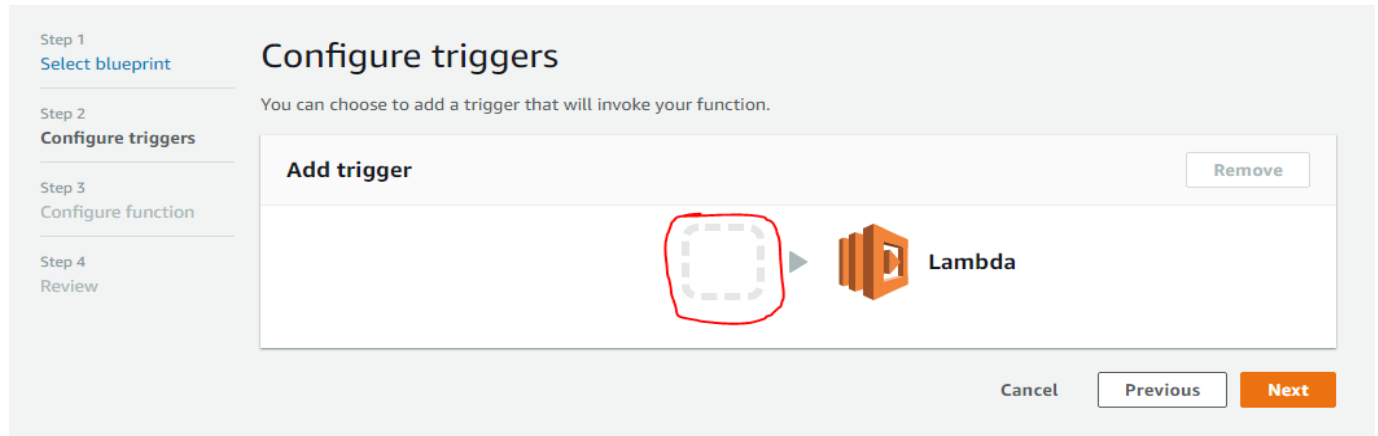
1. Make sure you've created an Amazon AWS Educate Account (see the instructions before). If you can't get this working please ask us.
2. Then visit this page: <https://goo.gl/JNhmbW>
(if it asks you to login, login with your AWS account details - these may be different from your Amazon Developer account)
3. Set your location (in the top right corner) to **EU (Ireland)**
4. Click on **Create function**:



5. Use the 'alexa-skill-kit-sdk-factskill' blueprint for nodejs6.10

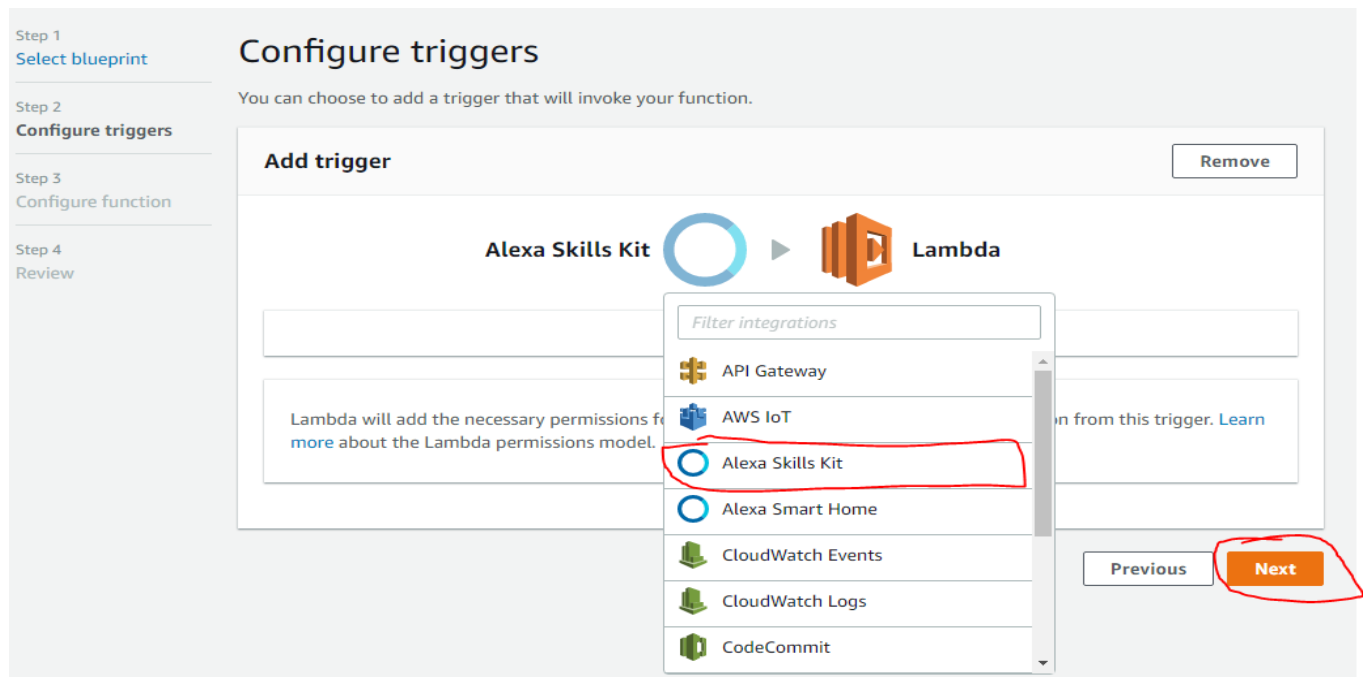


6. Click your mouse in the dotted square as shown



7. Select **Alexa Skills Kit**, and hit **Next**

If this is not available, make sure your location (in the top right corner) is set to **EU (Ireland)**



8. Choose a sensible name for your function (probably its name without the same), then paste the code you wrote before into the **Lambda function code**.

Step 1
Select blueprint

Step 2
Configure triggers

Step 3
Configure function

Step 4
Review

Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Basic information

Name*

Description

Runtime*

Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. [Learn more](#) about deploying Lambda functions.

Code entry type

```
1 'use strict';
2 console.log('Loading function');
3
4 /* Syslog format parser */
5 const parser = '^(?:\b(?:Jan(?:uary)?|Feb(?:ruary)?|Mar(?:ch)?|Apr(?:il)?|May|Jun(?:e)?|J
6   |Sep(?:ember)?|Oct(?:ober)?|Nov(?:ember)?|Dec(?:ember)?)\b\s+(?:0[1-9])|(?:[12]
7   '(?:2[0123]|01)?[0-9]):(?:[0-5][0-9]):(?:[0-5][0-9]|60)(?:[^\s,]*[0-9]+)?))' (
8   ')?(?:[a-zA-Z0-9._-]+) ([\w\._/%-]+)(?:\[(?:[1-9][0-9]*)\])?: (.*)';
9
10 exports.handler = (event, context, callback) => {
```

9. Don't change anything else, and scroll down to the bottom

Handler and role

Handler*

Role*

Existing role*

role permissions

Execution

Policy

Don't Allow

10. to control the code execution performance and costs for your Lambda function. Char

For **Role**, choose **create a custom role**

Then when the console pops up, select **Allow**

Finally, **existing role** choose **lambda_basic_execution**, then click next

Lambda function handler and role

Handler*
The module-name.export value in your function. For example, "index.handler" would call exports.handler in index.js.

Role*
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Existing role*
You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

► Tags

► Advanced settings

* These fields are required.

Cancel

11. Scroll down and click the orange **Create function** button (you're probably getting good at this now).

Lambda function handler and role

Handler
index.handler

Role
lambda_basic_execution

Advanced settings

Memory (MB)
128

Timeout (s)
7

VPC
No VPC

Enable active tracing
No

KMS key
(default) aws/lambda

Cancel

12. If you get warnings (triangles) you can ignore them. If you get errors however (big red box) please ask us for help!

13. Click the orange **Test** button.

Lambda > Functions > qwer

ARN - arn:aws:lambda:eu-west-1:609186337718:function:qwer

qwer

Qualifiers ▼ Actions ▼

14. You should get to this page. In the dropdown, pick **Alexa Start Session**, then click the orange **Save and test** button.

Input test event

Use the editor below to enter an event to test your function with. You can edit the event again by choosing **Configure test event** in the Actions list. Note that changes to the event will only be saved locally.

Sample event template

Alexa Start Session

Q

Alexa Start Session

Alexa Intent - Recipe

Alexa Smart Home - Turn Off

Alexa Intent - Answer

Alexa End Session

Alexa Smart Home - Control

Alexa Smart Home - Turn On

Alexa Intent - MyColorIs

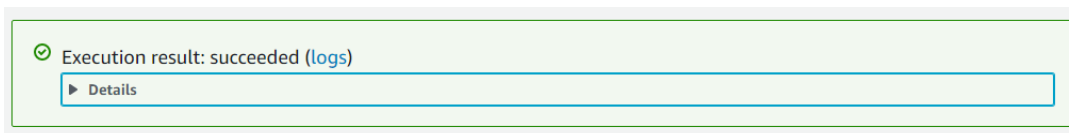
Lex

Lex - Make Appointment

```
20 ~ "context": {
21 ~   "AudioPlayer": {
22 ~     "playerActivity": "IDLE"
```

Cancel Save **Save and test**

15. If successful, you should get this message, if not put your hand up we will come round and help!



16. Next, look in the top right and copy the the text that comes after “ARN -”:

Lambda > Functions > qwer

ARN - arn:aws:lambda:eu-west-1:609186337718:function:qwer

Qualifiers Actions Test

✓ Congratulations! Your Lambda function "qwer" has been successfully created and configured with as a trigger in a disabled state. We recommend testing the function behavior before enabling the trigger.

⚠ This function contains external libraries. Uploading a new file will override these libraries.

Code Configuration **Triggers** Tags Monitoring

Alexa Skills Kit
alex-appkit.amazon.com

To configure your Alexa skill, go to the [Alexa Developer Portal](#).

+ Add trigger Refresh triggers

► View function policy

Delete

e.g. the above example it would be 'arn:aws:lambda:eu-west-1:60918...'

17. Go back to the Amazon Developers site at <https://goo.gl/DwfySL> and click Edit on your skill

18. Go to the configuration tab, select **AWS Lambda ARN** and paste in your arn. Don't change any of the other options. Then click **next**.

Global Fields
These fields apply to all languages supported by the skill.

Endpoint
Service Endpoint Type: ☒ **AWS Lambda ARN (Amazon Resource Name)** ⓘ ☐ HTTPS
Recommended
AWS Lambda is a server-less compute service that runs your code in response to events and automatically manages the underlying compute resources for you.
[More info about AWS Lambda](#)
[How to integrate AWS Lambda with Alexa](#)

Default
arn:aws:lambda:eu-west-1:338337944728:function:collegeScorecardEU

Provide geographical region endpoints? (Optional) ⓘ ☐ Yes ☒ No

Account Linking
Do you allow users to create an account or link to an existing account with you? ⓘ
[More info about Account Linking](#)
[Tips for successful Account Linking](#)

Permissions
Request users to access resources and capabilities
Please request permissions to resources and capabilities that are absolutely core to the customer experience delivered by the skill.
[Learn More](#)

☐ Device Address
☐ Full Address ⓘ
☐ Country & Postal Code Only ⓘ

☐ Lists Read ⓘ
☐ Lists Write ⓘ

See [Certification Requirements](#) in our technical documentation as you develop your skills and prepare to submit to Amazon.

Testing

1. Edit your skill the Amazon Developers site at <https://goo.gl/DwfySL>
2. In the bar on the left, click 'Test' (NB: You must have finished configuring you skill - see instructions above)
3. In the **Service Simulator**, try typing an **input utterance** (1). You'll then hopefully be able to see the **Lambda Response** (2) which should include your speech. If you want to hear how it'll sound you can press

the **Listen** (3) button at the bottom right (however you'll probably need headphones for this)

The screenshot shows the AWS Lambda console interface for a Text-based function. At the top, there are tabs for 'Text' and 'JSON'. Below them is an 'Enter Utterance' field containing the text 'hello'. To the right of this field is a red circle with the number '1' inside. Below the input field are two buttons: 'Ask College Scorecard' and 'Reset'. The main area is split into two panels: 'Lambda Request' and 'Lambda Response'. The 'Lambda Request' panel shows a JSON object with session, application, user, and request details. The 'Lambda Response' panel shows a JSON object with version, response, and outputSpeech details. In the 'outputSpeech' object, the 'ssml' field is highlighted with a red circle containing the number '2' and a red line pointing to its value: '<say-as> College Scorecard can get'. At the bottom right of the 'Lambda Response' panel, there is a 'Listen' button with a play icon, which is circled in red with the number '3' next to it.

Improving your code

If you want to edit your code, you can paste it back into a text editor. If you're not sure what to use you can try <http://requirebin.com/>

If you don't know JavaScript try deciphering what the code does. There's a good JavaScript quick tutorial at <http://jsforcats.com/> (ignore the stuff about 3rd party libraries).

Something that's quite fun to try out are Speechcons, simply put them in your speech response with the <say-as> tag: <https://goo.gl/njHqLV>

If you want to do things other than just tell your user stuff, you should check out <https://github.com/alexa/alexa-skills-kit-sdk-for-nodejs>

You can ask them for stuff (although you'll also have to handle their responses), tell them and send the Alexa app cards, and can save stuff into this.attributes so it's remembered between each time a user uses your skill.

If you're really far ahead check out Amazon's website for some samples: <https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/using-the-alexa-skills-kit-samples>

After making any changes you want to your code in your editor, copy it and once again replace the old lambda code with your new code, then press save and test.