How to create an Alexa Proactive Skill...

--IMPORTANT—

* Create a file which you will place everything you download into. Make sure you use this file only otherwise the parts get lost and do not link up correctly.
* Proactive Event Schemas…

[Proactive Events Schemas | Alexa Skills Kit (amazon.com)](https://developer.amazon.com/en-US/docs/alexa/smapi/schemas-for-proactive-events.html)

--STEP 1--

--Using Alexa Developer Console—

Create your skill.

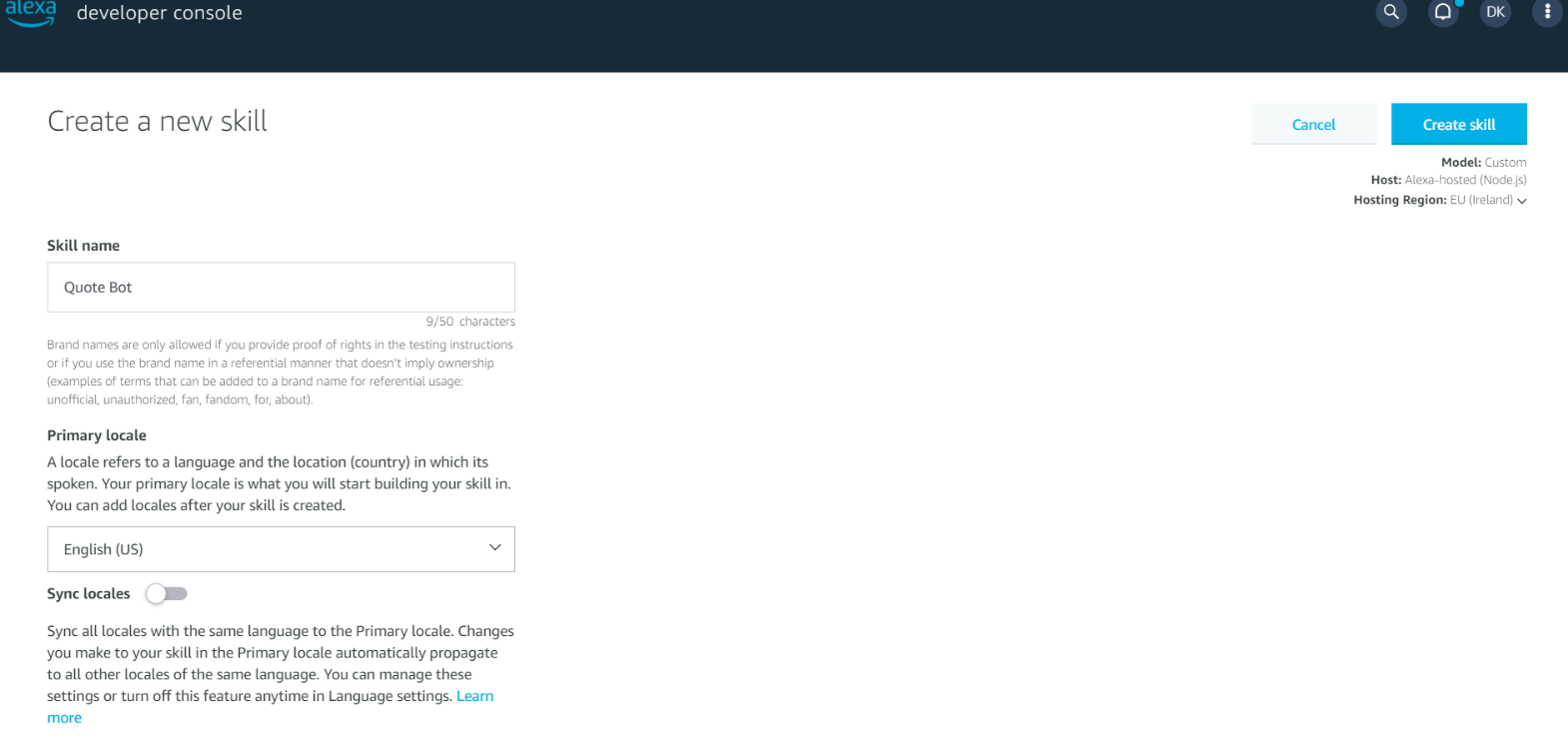
Give it a name and choose what model you want to apply to your skill. (I have chosen basic.)

Make your skill Alexa hosted.

Just under Create skill make sure you enable the correct Hosting Region.

Press create skill.

Pick a template for your skill and continue.

Your skill is now created.

--STEP2—

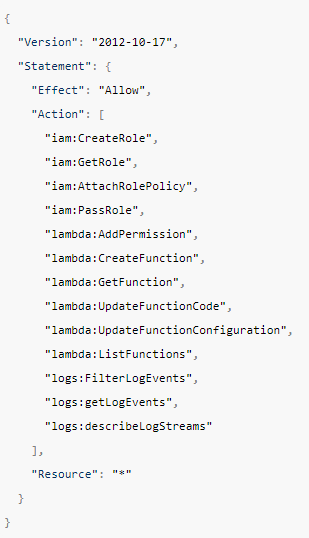
--Using AWS Management Console—

Type IAM into the search bar

On the left there should be a policies header, click into it.

Create a policy and switch the first screen from visual editor to JSON.

Delete the current JSON file and paste this file into it instead:



[Manage Credentials with ASK CLI | Alexa Skills Kit (amazon.com)](https://developer.amazon.com/en-US/docs/alexa/smapi/manage-credentials-with-ask-cli.html#create-aws-credentials)

Click next tags, then next:review.

Name your policy and create it.

Now, create a user for your policy

For the access type, pick programmatic.

Click next permissions and then “attach existing policies”

Filter the policies by customer managed and add the policy you just made.

Click next on the tags and create the user.

At the end you should see a success message and directly under it a “download.csv” button.

Click the button and save it to the important folder from earlier

--STEP 3—

--Creating a Lambda on AWS Management Console—

On AWS Click “Lambda” and then functions

Next, create a function.

You can author this from scratch.

Give it a name and choose an up to date runtime for the function.

Add a trigger to your Lambda.

Choose the Alexa skill kit option.

Go back to your Alexa developer console and go to build and then endpoint.

Here you will see your skill id.

Copy and paste it into your trigger.

You will find a copy arn button on the top right of your lambda home page.

Copy it into the default region under where you got your skill id.

--STEP 4—

--Setting up ASK CLI--

First, go into the code tab on the developer console and hit download.

Download your skill and then extract it to the important file.

Open up your computers terminal and go to your file using the cd command

Now, install “nodeJs” and “npm package manager” on your computer if you don’t already have them

After these are installed download ask-cli with this command: “npm install -g ask-cli”

Go to your “Users-> <user-name> -> .aws -> credentials” file on your computer.

Add this to your credentials file:

[<CLI\_USER\_NAME>] (Check .csv you downloaded earlier)

aws\_access\_key\_id=<aws\_access\_key\_id\_value>

aws\_secret\_access\_key=<aws\_secret\_access\_key\_value>

Type “ask init” into your terminal.

You are asked for:

your skill id (which is found in endpoint on the developer console)

your package path (click on you’re the skill file location and paste it into the cli)

exclude the lambda pack and then press enter.

Do not click the yes or no yet. Leave it there in the command line and come back to it.

Open the skill.json file you download in vsCode.

You must add permissions to this file to enable alexa notifications. You cannot do this on the developer console because the skill manifest file is not accessible.

To add notifications:

"permissions": [

     {

       "name": "alexa::devices:all:notifications:write"

     }

]

Also you must define the schema you choose here. In my case I added:

 "publications": [

        {

          "eventName": "AMAZON.MessageAlert.Activated"

        }

      ]

Next provide your Lamda ARN in your uri endpoints.

Now after this is all saved go back to the terminal and click yes.

--STEP 5—

--Deploying Alexa Skill with the CLI—

Type “ask deploy” into your command line.

Go back to your Amazon Developer Console and click Test.

Enable the development by turning it from “off” to “development”

Go onto the amazon web or phone app and go into settings.

Click notifications, turn them on for your skill and save the permissions.

If the skill.json has the correct permissions and you have enabled them on your account the proactive notifications will be ready to be performed.

--STEP 6—

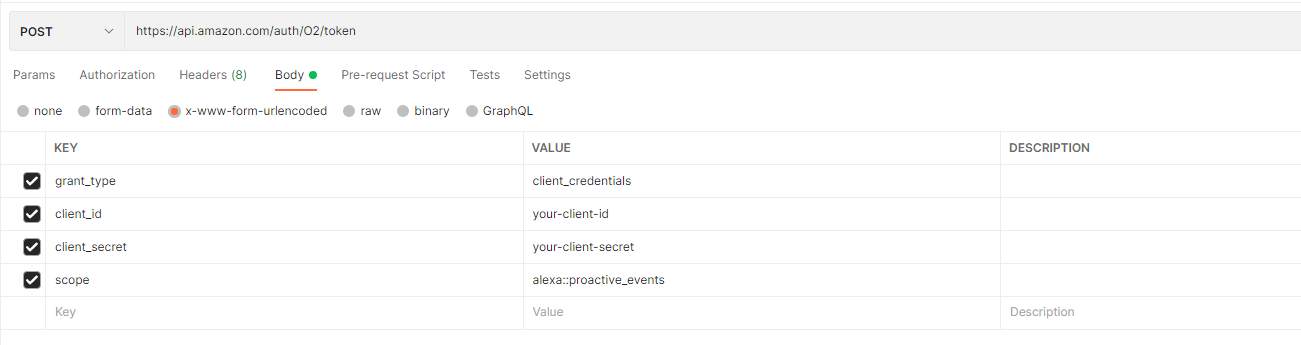
--Testing the notifications in Postman—

Log into Postman and enter your workspace.

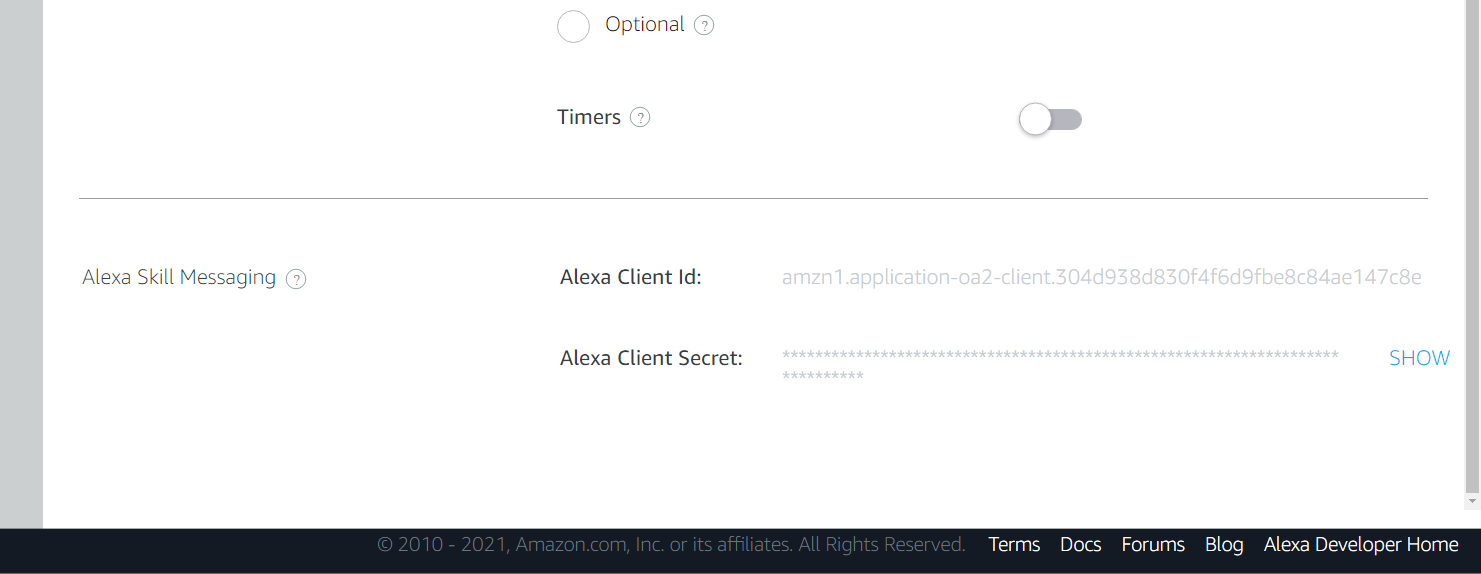
Press the “new” button on the left.

Change the GET Request to a POST Request and enter this url:

https://api.amazon.com/auth/O2/token

Click Params and enter the following parameters:

For the parameter values you get your client ID and Secret values from the permissions section at the end of the page.



Next, open the headers tab and add a Key called “Content-type” with the value of :

< application/x-www-form-urlencoded >

Send your request. If you have done everything right so far you get a Bearer Atc|token

Press the “new” button again and just like last time set the url to POST and paste this link in:

https://api.eu.amazonalexa.com/v1/proactiveEvents/stages/development (Europe)

Open Headers tab and add a key Content-Type with a value application/json

Add another key called “Authorization” and then in value put Bearer <your token from previous api>

\*IT IS IMPORTANT TO MAKE SURE THERE IS A SPACE BETWEEEN BEARER AND THE TOKEN\*

Go to the body part next and add you events json content (see link at the top of the page for schemas)

The bodys JSON schemas looks like this:



Note at the end of schema there is relevant audience. This is interchangeable between two different options, multicast and Unicast. Here is the difference:

Multicast – Sends a notification to every device that has the skill activated

Unicast – Sends a notification to a specific device who has your skill activated.

If you choose this option a userId is required as a parameter.

"relevantAudience": {

        "type": "Unicast",

        "payload": {

            "user": "your user-id"

        }

If you test your skill in the developer console a JSON file comes up on the right. If you go through this you will see a user-id come up. Copy and paste this into your Postman body and it will send the notification to a certain users account/device.

Hit Send and if you get a 202 accepted notification your Alexa should turn yellow. Otherwise you must identify your error and try fix it. Follow the error codes until you solve them and use this page to help you: [Proactive Events API Reference | Alexa Skills Kit (amazon.com)](https://developer.amazon.com/en-US/docs/alexa/smapi/proactive-events-api.html)