

Create your own „Alexa Domoticz Smart Home Skill“

Prerequisites:

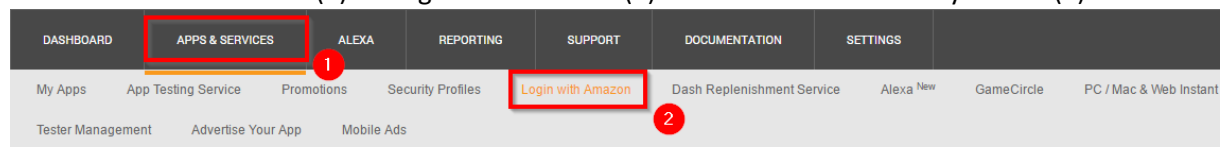
- Register Account: <https://developer.amazon.com>
- Register Account: <https://console.aws.amazon.com>
- External connect from the internet to domoticz (port forwarding, dns name)

Create oauth2 authorization provider:

Alexa smart home skill requires an OAuth2 authorization.

1.1 Sign in: <https://developer.amazon.com>

1.2 Click to APPS & SERVICES (1) → Login with Amazon (2) → Create a New Security Profile (3)



Login with Amazon

Login with Amazon allows users to login to registered third party websites or apps ('clients') using their Amazon user name and password. Clients may ask the user to share some personal information from their Amazon profile, including name, email address, and zip code. To get started, select an existing Security Profile or create a new Security Profile. [Learn More](#)



1.3 Type in any Security Profile Name, Description and Privacy URL, (Optional Logo Image)
Click to Save

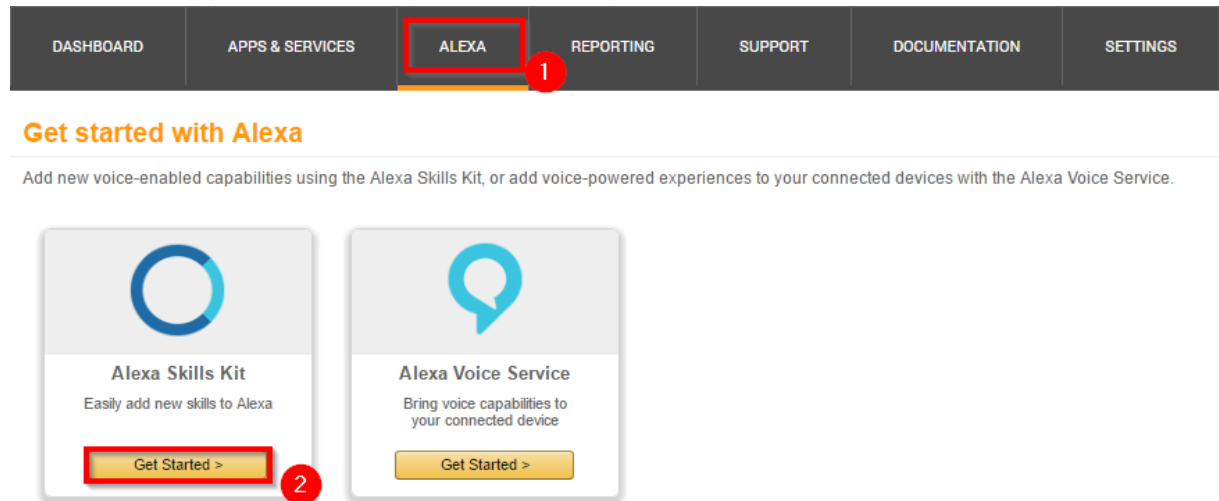
1.4 Click to “Show Client ID and Client Secret” and save Client ID and Client Secret to notepad

1.5 Do not close this page, one last step is needed at the end

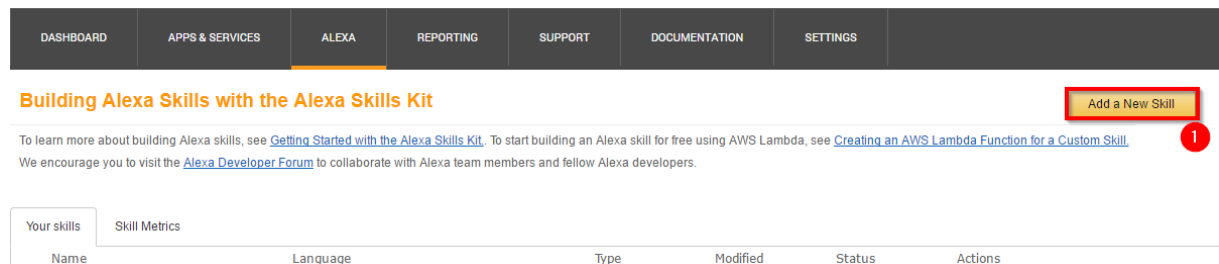
Create alexa skill:

2.1 Sign in: <https://developer.amazon.com>

2.2 Click to ALEXA (1) → Get Started (2, Alexa Skills Kit)



2.3 Click to “Add a New Skill”



2.4 Select Skill Type → “Smart Home Skill API” (1)

Choose your skill language (2)

Input the name of your skill → For example “Domoticz” (3)

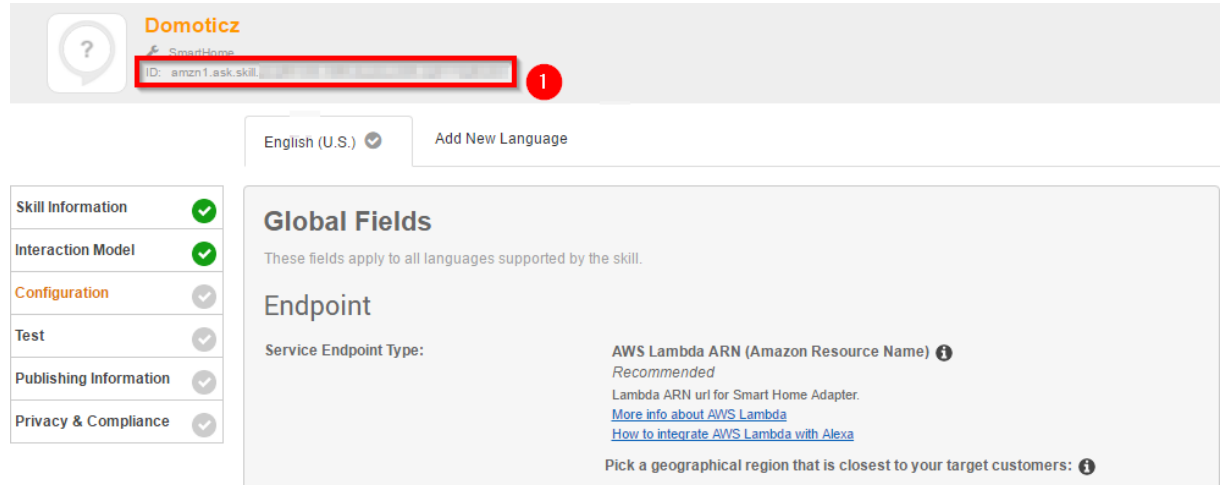
Click to Next (4)

The screenshot shows the 'Create a New Alexa Skill' form. On the left is a sidebar with sections: Skill Information, Interaction Model, Configuration, SSL Certificate, Test, Publishing Information, and Privacy & Compliance, each with a checkmark. The main form has the following fields: 'Skill Type' with radio buttons for 'Custom Interaction Model', 'Smart Home Skill API' (selected and highlighted with a red box and a red circle with '1'), and 'Flash Briefing Skill API'; 'Language' with a dropdown menu showing 'English (U.S.)' (highlighted with a red box and a red circle with '2'); 'Name' with a text input field containing 'Domoticz' (highlighted with a red box and a red circle with '3'); and a 'Next' button (highlighted with a red box and a red circle with '4') at the bottom right.

2.5 In section “Interaction Model” click to Next

Version 1.2

2.6 Copy Application ID to notepad



The screenshot shows the Alexa Developer Console for a skill named "Domoticz". At the top, there is a header bar with the skill name and a "SmartHome" icon. Below this, the "ID" field is highlighted with a red box and a red circle with the number "1". The ID is "amzn1.ask.skill.". Below the header, there is a language selector showing "English (U.S.)" and an "Add New Language" button. On the left, there is a sidebar with a list of sections: "Skill Information", "Interaction Model", "Configuration", "Test", "Publishing Information", and "Privacy & Compliance". The "Configuration" section is currently selected. The main content area is titled "Global Fields" and contains the "Endpoint" section. The "Endpoint" section has a "Service Endpoint Type" dropdown set to "AWS Lambda ARN (Amazon Resource Name)". Below this, there is a "Recommended" section with a "Lambda ARN url for Smart Home Adapter" and two links: "More info about AWS Lambda" and "How to integrate AWS Lambda with Alexa". At the bottom, there is a prompt to "Pick a geographical region that is closest to your target customers".

2.7 Now you have to create the lambda service to continue Do not close this website

[Create lambda service:](#)

Alexa smart home skill requires an OAuth2 authorization.

3.1 Sign in: <https://console.aws.amazon.com>

3.2 Click to Lambda in "All services" and then to "Get Started Now"

3.3 On the right top corner select your region

EU Language → Recommended "Ireland"

US Region → Recommended "N.Virginia"

The smart home skill runs from Ireland and N.Virginia. In the future, the other servers should work too.

If your skill is in german, you have to choose "Ireland".

3.4 Choose "Blank Function" (1)

Select blueprint



Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under [CC0](#).

Select runtime ▼

Filter

« < Viewing 1-9 of 90 > »

Blank Function Configure your function from scratch. Define the trigger and deploy your code by stepping through our wizard. custom	kinesis-firehose-syslog-to-json An Amazon Kinesis Firehose stream processor that converts input records from RFC3164 Syslog format to JSON. nodejs · kinesis-firehose	alexa-skill-kit-sdk-factskill Demonstrate a basic fact skill built with the ASK NodeJS SDK nodejs · alexa
batch-get-job-python27 Returns the current status of an AWS Batch Job. python2.7 · batch	kinesis-firehose-apachelog-to-j... An Amazon Kinesis Firehose stream processor that converts input records from Apache Common Log format to python2.7 · kinesis-firehose	cloudfront-modify-response-he... Blueprint for modifying CloudFront response header implemented in NodeJS. nodejs · cloudfront · response head...
s3-get-object-python An Amazon S3 trigger that retrieves metadata for the object that has been updated. python2.7 · s3	config-rule-change-triggered An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types. nodejs4.3 · config	lex-book-trip-python Book details of a visit, using Amazon Lex to perform natural language understanding python2.7 · lex

3.5 Click to empty border (1) and select “Alexa Smart Home” (2)

Configure triggers

You can choose to add a trigger that will invoke your function.

Remove

Filter integrations

- API Gateway
- AWS IoT
- Alexa Skills Kit
- Alexa Smart Home
- CloudFront
- CloudWatch Events - Schedule
- CloudWatch Logs
- CodeCommit

Cancel Previous Next

3.6 Input Application ID (1) from notepad and enable Trigger (2)

Click to Next (3)

Configure triggers

You can choose to add a trigger that will invoke your function.

Remove

Alexa Smart Home Lambda

Application Id

Choose **Submit** will create a resource policy that allows the Amazon Alexa service to call your Lambda function. To configure the Alexa service to work with your Lambda function, go to the [Alexa Developer](#) portal. [Learn more](#) about the Lambda permission model.

Enable trigger ☒

Cancel Previous Next

3.7 Set a name and description

Set the Runtime* to "Node.js 4.3"

Configure function

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

Name*	<input type="text" value="domapi"/>
Description	<input type="text" value="Alexa domoticz smart home skill framework"/>
Runtime*	<input type="text" value="Node.js 4.3"/>

3.8 Download all files from https://github.com/madgeni/alexa_domo and extract zip file

Open example_conf.json and input your values.

Host should be your external address (DNS name or IP address)

Save the changed file with filename conf.json

3.9 Zip files conf.json, domapi.js, package.json and folder node_modules together



3.10 Select Upload a .ZIP file (1)

Click to Upload and select your ZIP file (2)

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	<input type="text" value="Upload a .ZIP file"/>	1
Function package*	<input type="button" value="Upload"/>	2

For files larger than 10 MB, consider uploading via S3.

- 3.11 Change the handler (1) to domapi.handler (or if you change the domapi.js filename, then to whatever you change it to)
Select "Create a custom role" (2)

Lambda function handler and role

Handler* 1

Role* ▼ 2

Existing role*

Advanced settings

- 3.12 Click to Allow

▼ Hide Details

Role Summary ?

Role Lambda execution role permissions

Description

IAM Role

Role Name

► View Policy Document

Don't Allow

Allow

- 3.13 Now the window should look like this

Click to Next

Lambda function handler and role

Handler* 1

Role* ▼ existing 1

Existing role* 1

Advanced settings

* These fields are required.

Cancel Previous **Next** 2

- 3.14 Click to Create function

3.15 Copy ARN address to notepad (1)

[Lambda](#) > [Functions](#) > domapi

ARN - arn:aws:lambda:eu-west-1:123456789012:lambda:function:domapi

Qualifiers ▾

Test

Actions ▾

1

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

Code

Configuration

Triggers

Monitoring

?



Alexa Smart Home

Application Id: amzn1.ask.skill.12345678901234567890123456789012

Disable

Delete

+ Add trigger

► View function policy

[Continue with Create alexa skill:](#)

4.1 Back to the section “Configuration” from step 2.6

Select your Region (1) and input your ARN address (2) from notepad or lambda website

Global Fields

These fields apply to all languages supported by the skill.

Endpoint

Service Endpoint Type:

AWS Lambda ARN (Amazon Resource Name) ⓘ

Recommended

Lambda ARN url for Smart Home Adapter.

[More info about AWS Lambda](#)

[How to integrate AWS Lambda with Alexa](#)

Pick a geographical region that is closest to your target customers: ⓘ

☐ North America ☒ Europe 1

Europe

arn:aws:lambda:eu-west-1: 2

4.2 Set “Authorization URL” (1)

https://www.amazon.com/ap/oa/?redirect_url=

and then copy the Redirect URL from further down the page and append it to the end of the Authorization URL

For example:

https://www.amazon.com/ap/oa/?redirect_url=https://layla.amazon.com/api/skill/link/xxxxxx

Set your “Client Id” from notepad or step 1.4. (2)

Set “Scope” (3)

profile:user_id

This will give your Alexa Skill access to a minimal amount of information about you from Amazon, in this case just your user_id

Select “Auth Code Grant” (4)

Set “Access Token URI” (5)

<https://api.amazon.com/auth/o2/token>

Set “Client Secret” from notepad or step 1.4 (6)

Set any “Privacy Policy URL” (7)

Click to Next

Account Linking

Authorization URL
The url where customers will be redirected in the companion app to enter login credentials.

https://www.amazon.com/ap/oa/?redirect_url=https://layla.amazon.com/api/skill/link/

Client Id
Unique public string used to identify the client requesting for authentication.

amzn1.application-oa2-client

Domain List (Optional)
The list of domains that the authorization URL will fetch content from. You can provide up to 15 domains.

[Add domain +](#)

Scope
List of permissions to request from the skill user. You can provide up to 15 scopes.

[Add scope +](#)

1 profile:user_id

Redirect URLs (Optional)
The list of valid HTTPS redirection endpoints that could be requested during authorization to redirect the user back to after the authorization process.
[Learn more.](#)

<https://layla.amazon.com/api/skill/link/>
<https://pitangui.amazon.com/api/skill/link/>

Authorization Grant Type (Optional)
Specifies the OAuth authorization grant that Alexa uses to obtain an access token from your provider.
[Learn more.](#)

☐ Implicit Grant ☒ Auth Code Grant

Access Token URI
This URI will be used for both access token and token refresh requests.

<https://api.amazon.com/auth/o2/token>

Client Secret

Client Authentication Scheme (Optional)

HTTP Basic (Recommended)

Privacy Policy URL
Link to the Privacy Policy for this skill. This is mandatory for account linking.

<https://developer.amazon.com>

4.3 Select Yes in “Start testing this skill” and click to Next

4.4 (Optional) Type in some informations and select images
Click to Save

[Link account back to your skill:](#)

5.1 Sign in: <https://developer.amazon.com> or back to the first step website (OAuth2)

5.2 Click to APPS & SERVICES → Login with Amazon

5.3 Click to Manage → Web Settings

5.4 Add the two ReDirect URLs from Step 4.2

Example: <https://layla.amazon.com/api/skill/link/xxxxxxxxxx>

<https://pitangui.amazon.com/api/skill/link/xxxxxxxxxx>

[Activate smart home skill in alexa:](#)

6.1 Open alexa app or alexa website (<https://alexa.amazon.com>)

6.2 Click to skills and then to Your skills

6.3 Select domoticz skill and click to activate

6.4 Login with your amazon credentials

6.5 Search devices in smart home

Additional Informations:


It's possible to set a device name only for alexa. In some cases, this is needed because alexa does not understand the name, the name is too long or too complicated.

Write in the "Description:" field (2) → Alexa_Name: Your Name for Alexa

So Device Name "Light Test" (1) will be ignored from alexa and the new name is "Your Name For Alexa".

Name: **Light Test** 1

Switch Type: On/Off

Switch Icon:  Light/Switch
A Lamp or Switch

On Delay: 0 (Seconds) 0 = Disabled

Off Delay: 0 (Seconds) 0 = Disabled

On Action: (Show)

Off Action: (Show)

Protected: ☐

Description: **Alexa_Name:Your Name For Alexa** 2

Save Delete

Appendix – TESTING via Lambda:

Under the Actions dropdown:

Configure test event allows you to send requests to your domoticz and interact with it, without having to install the smart skill on your alexa.

Here are a 3 – 1 discovery, 1 for lights, 1 for scenes:

Just copy one in, change the applicationId to a Domoticz IDX and test it.

```
{
  "header": {
    "messageId": "01ebf625-0b89-4c4d-b3aa-32340e894688",
    "name": "TurnOffRequest",
    "namespace": "Alexa.ConnectedHome.Control",
    "payloadVersion": "2"
  },
  "payload": {
    "accessToken": "123",
    "appliance": {
      "additionalApplianceDetails": {
        "switchis": "On/Off",
        "WhatAmI": "light"
      },
      "applianceId": "CHANGETHISTOANIDXINYOURDOMOTICZ"
    }
  }
}
```

```
{
  "header": {
    "messageId": "01ebf625-0b89-4c4d-b3aa-32340e894688",
    "name": "TurnOffRequest",
    "namespace": "Alexa.ConnectedHome.Control",
```

Version 1.2

```
"payloadVersion": "2"
},
"payload": {
  "accessToken": "123",
  "appliance": {
    "additionalApplianceDetails": {
      "WhatAmI": "scene"
    },
    "applianceId": CHANGETHISTOANIDXINYOURDOMOTICZ
  }
}

{
  "header": {
    "messageId": "6d6d6e14-8aee-473e-8c24-0d31ff9c17a2",
    "name": "DiscoverAppliancesRequest",
    "namespace": "Alexa.ConnectedHome.Discovery",
    "payloadVersion": "2"
  },
  "payload": {
    "accessToken": "123"
  }
}
```