# Create your Smart Home Skill:

<http://developer.amazon.com> - Sign in using your normal Amazon account:

At the top, you’ll see Dashboard/Apps & Services/Alexa/Reporting/Support/Documentation/Settings

Select Alexa, then Alexa Skills Kit, then Add a New Skill.

On the Create a New Skill page – select Smart Home Skill, give it a name, Save, then Next



It will then create your Smart Home Skill. Before continuing here, you will need to create the Lambda function. (This Lambda function is where the code actually runs).

# create your Lambda function:

<https://console.aws.amazon.com>

**NB – there are several different regions where these Lambda functions run. Alexa skills should run from both N.Virginia & Ireland, however I’ve only had success from N.Virginia**



Select Lambda, then in the next page, Create A Lambda Function.

Type ‘alexa’ in the Select Blueprint search field and select ‘alexa-smart-home-skill-adapter’



The first thing you’ll do is to tie it back to your Smart Home Skill.

Back in your Smart Home Skill, you need the Application ID:



Copy this, and paste it into the ApplicationID field on Configure Triggers, enable it, then press Next.

Now – go to <https://github.com/madgeni/alexa_domo>

Download: domapi.js, package.json, example\_conf.json & the node\_modules folder. Firstly, rename the example\_conf.json to conf.json, and edit it. You will want to put your externally facing IP or DNS name in the hostname field, port, username and password in there. Then zip these files together.

Back to Lambda:



Give it a name and description, leave the runtime as Node.js, then upload that zip file.



Change the handler name to domapi.handler (or if you change the domapi.js filename, then to whatever you change it to).



For the Role \* - select ‘Choose an existing role’, then for Existing role, select ‘lambda\_basic\_execution’ role.

This whole piece is simply setting up the conditions for the lambda instance that runs the code, rather than interacting with your domoticz.

## Review – click Create Function.

Back to the Smart Home Skill:

Alexa requires oauth2 to interact, however Domoticz doesn’t provide an oauth2 token, so you have two choices – link the skill to your Amazon account, or use a spoof oauth2 provider.

Spoof details here:

This is a fake OAUTH server, which allows you to use this as a workaround:

Account Linking [b]Authorization URL:[/b] <https://ve0kyj5tp5.execute-api.us-east-1.amazonaws.com/test/HomeSkillAuthorization>

Client id: alexa - any value is fine, since it's ignored

Domain List: Ignore this

Scope: alexa - any value is fine, since it's ignored

Access Token URI:[/b] <https://ve0kyj5tp5.execute-api.us-east-1.amazonaws.com/test/OAuthAccessToken>

Client Secret: alexa - any value is fine, since it's ignored

Privacy Policy: https://ve0kyj5tp5.execute-api.us-east-1.amazonaws.com/test/PrivacyPolicy - any value is fine).

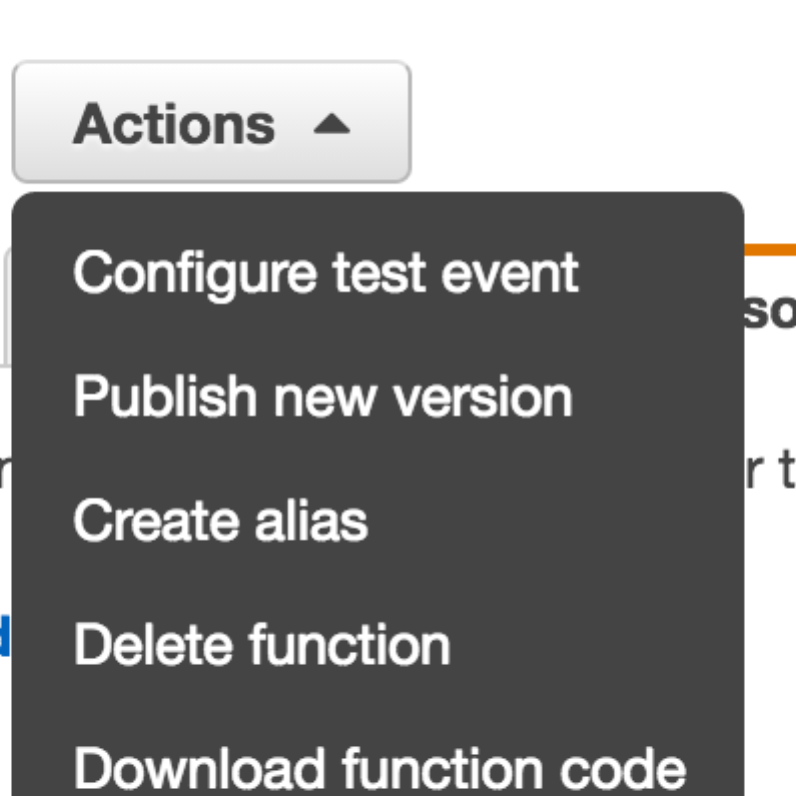
Click **SAVE**

That should be it!

<http://alexa.amazon.com> - skills. At the top right, you should see ‘your skills’. Click on that, and you should see your skill. Enable it, and then discover your devices!

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",

"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"

}

}

OAUTH –

So Alexa requires OAUTH2, but it’s a pain, and if the alexa app is for your own consumption, unnecessary!

This is a fake OAUTH server, which allows you to use this as a workaround:

Account Linking [b]Authorization URL:[/b] <https://ve0kyj5tp5.execute-api.us-east-1.amazonaws.com/test/HomeSkillAuthorization>

Client id: alexa - any value is fine, since it's ignored

Domain List: Ignore this

Scope: alexa - any value is fine, since it's ignored

Access Token URI:[/b] <https://ve0kyj5tp5.execute-api.us-east-1.amazonaws.com/test/OAuthAccessToken>

Client Secret: alexa - any value is fine, since it's ignored

Privacy Policy: https://ve0kyj5tp5.execute-api.us-east-1.amazonaws.com/test/PrivacyPolicy - any value is fine).

Click **SAVE**