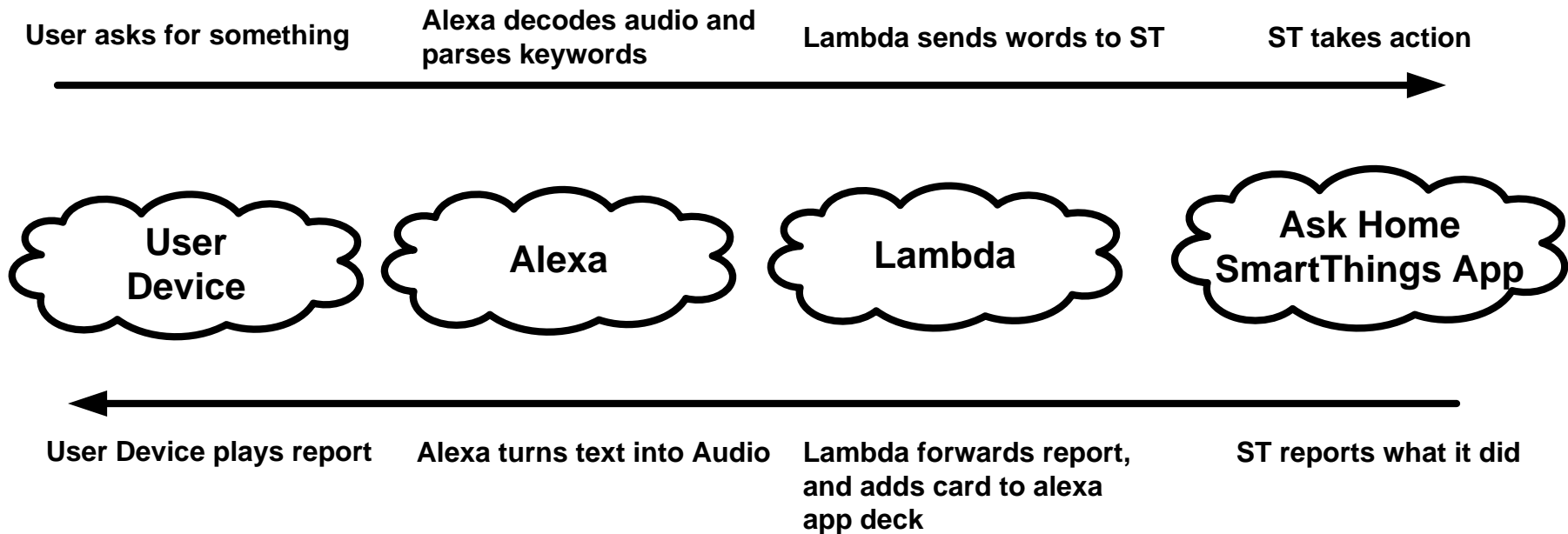


Ask Home Overview



The Ask Home design intention was to have SmartThings do as much as possible, so that SmartThings programmers would not have to make changes to code anywhere else. The exception to that is in the Alexa Skill Set, she really needs to know key words and phrase formats. The Lambda code doesn't really need to be changed, but the Lambda Test environment can help you debug a smart things app problem (along with the smarthings log) because it's the only thing that talks directly to SmartThings. Testing at Alexa helps with making sure alexa is pulling out the right keywords for the phrases you type in. And testing on the user device helps with any problems with word recognition.

The Ask Home Smart App is meant for programmer customization...it will be non functional for an end user until a programmer has customized it for the users nouns and devices. To make it work, apply for an Amazon Developer Account, use the screen shots and cut-paste lists to fill in the blanks in Alexa and Lambda, and put the SmartThings AskHome app in the Smart Things Apps. After that it is 1,2,3. 1. Figure out what nouns you will use, 2. Add device inputs at the top of the code for the smart things devices you need, and 3. put the nouns and subroutine calls in the switch/case table. There are pre-constructed capability handling routines which you can layer for each noun to perform a wide Variety of actions. Look at the SmartApp code to see the examples.



AskHome
DEVELOPMENT
4/28/16

[Getting started](#)

*Fields required for certification

Skill Information	✓
Interaction Model	✓
Configuration	✓
Test	✓
Publishing Information	✓
Privacy & Compliance	✓

Application Id
The ID for this skill
amzn1.echo...K-ams.am...ed4991...f-49c5-b...cca66a7f

Skill Type *
You can choose a Skill API or define the interaction model. [Learn more](#)
☒ Custom Interaction Model
☐ Smart Home Skill API

Name *
The name of this skill. This is the name displayed in the Alexa App.
AskHome

Invocation Name *
The name users will say to interact with this skill. This does not have to be the same as the skill name. The invocation name must comply with the [Invocation Name Guidelines](#)
home

Save

Submit for Certification

Next

Skill Information	✓
Interaction Model	✓
Configuration	✓
Test	✓
Publishing Information	✓
Privacy & Compliance	✓

Intent Schema*

The schema of user intents in JSON format. For more information, see [Intent Schema](#). Also see [built-in slots](#) and [built-in intents](#).

```
1 {  
2   "intents": [  
3     {  
4       "intent": "Home",  
5       "slots": [  
6         {  
7           "name": "Operator",  
8           "type": "LIST_OF_OPERATORS"  
9         },  
10      {  
11        "name": "Noun",  
12        "type": "LIST_OF_NOUNS"  
13      },  
14      {  
15        "name": "Operand",  
16        "type": "LIST_OF_OPERANDS"  
17      }  
18    ]  
19  }  
20 }
```

Custom Slot Types

Custom slot types to be referenced by the Intent Schema and Sample Utterances

For general information about custom slots, see [Custom Slot Types](#).

Example: TOPPINGS - cheese | onions | ham (note: newlines displayed as | for brevity)

Type	Values	
LIST_OF_INQUISIT...	is are who what when where why how which	Edit
LIST_OF_NOUNS	Bedroom Light Medic Button Water Shutoff Valve Kitchen Air Baseme...	Edit
LIST_OF_OPERANDS	black white blue green yellow orange purple pink red 0 1 ...	Edit
LIST_OF_OPERATORS	status on off presence present not present open close closed ...	Edit

Sample Utterances*

Phrases end users say to interact with the skill. For better results, provide as many samples as you can. Note that you must select three of these to use as your Example Phrases on the Description tab.

For more information, see [Sample Utterances](#).

```
1 Home about the {Noun}  
2 Home if the {Noun} are {Operator}  
3 Home {Inquisitor} {Noun} are {Operator}  
4 Home {Inquisitor} {Operator} is the {Noun}  
5 Home {Inquisitor} the {Noun} {Operator}  
6 Home {Inquisitor} the {Noun} is {Operator}  
7 Home {Inquisitor} the {Noun} is turned {Operator}  
8 Home for the {Operator} of the {Noun}  
9 Home for the {Noun} {Operator}  
10 Home for the {Operator} of {Noun}  
11 Home to turn {Operator} the {Noun}  
12 Home to turn the {Noun} {Operator}  
13 Home {Noun} {Operator}  
14 Home to {Operator} {Noun}  
15 Home {Inquisitor} {Noun} {Operator}
```

*Fields required for certification

Skill Information	✓
Interaction Model	✓
Configuration	✓
Test	✓
Publishing Information	✓
Privacy & Compliance	✓

Endpoint *

The URL for the service endpoint, e.g.
<https://myskill.ishere.com/somepath>,
or the Lambda ARN.
[More info about AWS Lambda](#)
[How to integrate AWS Lambda with Alexa](#)

☐ HTTPS ☒ Lambda ARN (Amazon Resource Name) ?

arn:aws:lambda:us-east-1:123456789012:function:SmartLink-MultiInter

Get this from Lambda!

Account Linking

Do you allow users to create an account
or link to an existing account with you?
[Learn more *](#)

☐ Yes ☒ No

Linking is not recommend for AskHome
But the clues are here for proper configuration
if you write your own app to use the Lambda core

Authorization URL *

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

https://graph.api.smartthings.com/oauth/authorize?response_type=code

Client Id *

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

76274 4a-4726-8 f3baea

smartthings.com

amazonaws.com

Domain List

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

Redirect URL

HTTPs redirection endpoint uri you want to
direct to after completing the authorization
interaction with user.

<https://pitangui.amazon.com/api/skill/link/3TJ1HT/6>

Authorization Grant Type

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 *

<https://graph.api.smartthings.com/oauth/token>

Client Secret *

Client Authentication Scheme

HTTP Basic (Recommended) ▾

AWS Services Edit

Lambda > Functions > SmartLink-MultiIntent

ARN - arn:aws:lambda:us-east-1:113011301130:function:Smr

Test Actions

Cut and paste this into the Alexa Developer Configuration page

Code Configuration Event sources API endpoints Monitoring

Code entry type ☒ Edit code inline ☐ Upload a .ZIP file ☐ Upload a file from Amazon S3

```
1 // Copyright 2016 Keith DeLong (n8xd)
2 // Apache 2.0 License - http://www.apache.org/licenses/LICENSE-2.0
3 'use strict';
4 exports.handler = function( event, context ) {
5     var https = require( 'https' );
6
7     var STappID = 'xxxxxx'; // AppID from Apps Editor
8     var STtoken = 'xxxxxx'; //Token from Apps Editor
9     //var STappID = 'xxxxxx'; // ID for Auto OAUTH
10    //if (event.session.user.accessToken) {STtoken = event.session.user.accessToken; }
11
12    if (event.request.intent.name == "Home") {
13        var Operator = event.request.intent.slots.Operator.value;
14        var Noun = event.request.intent.slots.Noun.value;
15        var Operand = event.request.intent.slots.Operand.value;
16        var Inquisitor = event.request.intent.slots.Inquisitor.value;
17        if (!Operator) {Operator = "none";}
18        if (!Noun) {Noun = "none";}
19        if (!Operand) {Operand = "none";}
20        if (!Inquisitor) {Inquisitor = "none";}
21        var url = 'https://graph.api.smarththings.com/api/smartapps/installations/' + STappID + '/' +
22            Noun + '/' + Operator + '/' + Operand + '/' + Inquisitor + '?access_token=' + STtoken;
23        console.log(url)
24        https.get( url, function( response ) {
25            response.on( 'data', function( data ) {
26                var resJSON = JSON.parse(data);
27                var speechText = 'The App on SmartThings did not return any message.'
28                if (resJSON.talk2me) { speechText = resJSON.talk2me; }
29                console.log(speechText);
30                output(speechText, context);
31                console.log("after the fact");
32            });
33        });
34    }
35}
```

Get the node.js code from github and paste it in here.

Go to your smart app on the Smarththings Website, and open it up in the editor. Do the Location/Simulation thing on the right side and get the appID and token. Paste those numbers into the xxxxxx at the top of the code.

Copy the ARN at the top right of the Lambda screen, and take that back to the Alexa Skills Kit and paste that into the slot at the top of the configuration page. So that the Skills kit knows where to send what it hears.

Under the Event sources tab here in Lambda, add the Alexa Skill Kit has permission to talk to it.