

# Building Spoken Language Apps for Amazon Echo and Alexa

---



**PETER HEINRICH**  
DEVELOPER EVANGELIST, AMAZON



@PeterDotGames



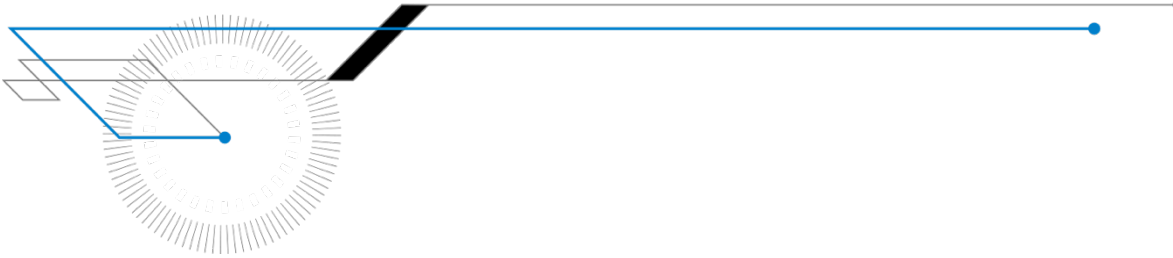
heinrich@amazon.com



**<http://developer.amazon.com/ask>**

**<http://developer.amazon.com/blog>**

# What is Alexa?





Alexa, Hello.

Skills are how you, as a developer, make Alexa smarter.

They give customers new experiences.

They're the voice-first apps for Alexa.

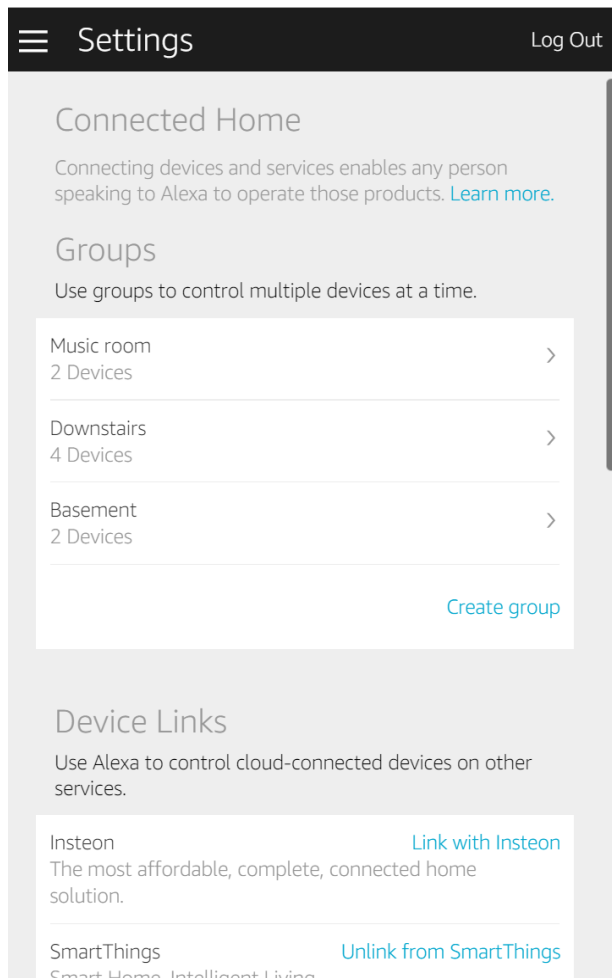
# The **Alexa** Platform



# Connected Home (CoHo) and Lighting API

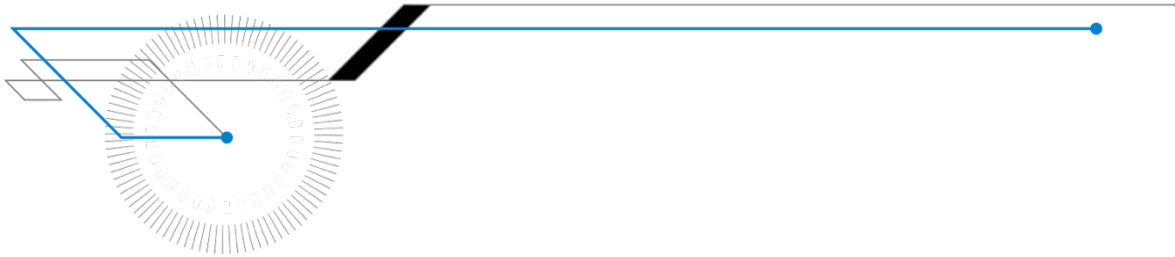


# Alexa App



<http://alexa.amazon.com>

# Alexa Skills Kit (ASK) Overview



# ALEXA SKILLS KIT (ASK)

<https://developer.amazon.com/ask>



## Alexa Skills Kit (ASK) Developer Preview

A free SDK that lets you easily add new voice capabilities

Alexa, the voice service that powers Echo, provides a set of built-in abilities, or skills, that enable customers to interact with devices in a more intuitive way using voice. Examples of these skills include the ability to play music, answer general questions, set an alarm or timer and more. With the Alexa Skills Kit, you can easily build and add your own skills to Alexa. Customers can access these new skills simply by asking Alexa a question or making a command.

### Quickly Build Skills with the Alexa Skills Kit (ASK)

The Alexa Skills Kit is a collection of self-service APIs, tools, documentation and code samples that make it fast and easy for you to add skills to Alexa. Using ASK, you can teach new skills to Alexa in just a few hours. No prior experience with speech recognition or natural language understanding is required. All of the code runs in the cloud — nothing is installed on any user



Get Started

Alexa Voice  
Service

Alexa Fund

#### GETTING STARTED

- [Getting Started with the Alexa Skills Kit](#)

# ALEXA VOICE SERVICE (AVS)

<https://developer.amazon.com/avs>



## Alexa Voice Service (AVS) Developer Preview

Bring voice capabilities to your connected device

### Introducing the Alexa Voice Service (AVS)

If you're a hardware maker and your connected device has a microphone and a speaker, the new Alexa Voice Service (AVS) [developer preview](#) enables you to add voice-powered experiences to your connected devices. Your customers can simply speak to Alexa through the microphone on your device and Alexa will respond through your device's speakers. This gives your customers access to Alexa's skills and capabilities, including built-in skills and those created by your or other developers using the [Alexa Skills Kit](#) (ASK). Examples of Alexa skills include the ability to answer general knowledge questions, provide weather forecasts, query Wikipedia and much more.



Get Started

[Alexa Skills Kit](#)

[Alexa Fund](#)

#### GETTING STARTED

- [Getting Started with the Alexa Voice Service](#)

# THE ALEXA FUND

<https://developer.amazon.com/alexafund>



## The Alexa Fund

\$100 million in investment to fuel voice technology innovation

### What is the Alexa Fund?

Experiences designed around the human voice will fundamentally improve the way people use technology. Since introducing Amazon Echo, we've heard from developers, manufacturers, and start-ups of all sizes who want to innovate with voice. The Alexa Fund--named for Alexa, the cloud-based voice service that powers Amazon Echo--provides up to \$100 million in investments to fuel voice technology innovation. Whether that's creating new Alexa capabilities with the [Alexa Skills Kit \(ASK\)](#), building devices that use Alexa for new and novel voice experiences using the [Alexa Voice Service \(AVS\)](#), or something else entirely, if you have a visionary idea, we'd love to talk to you.



Alexa Voice  
Service

Apply to the fund

Alexa Skills Kit












GETTING STARTED

■ [Tell us about your idea](#)

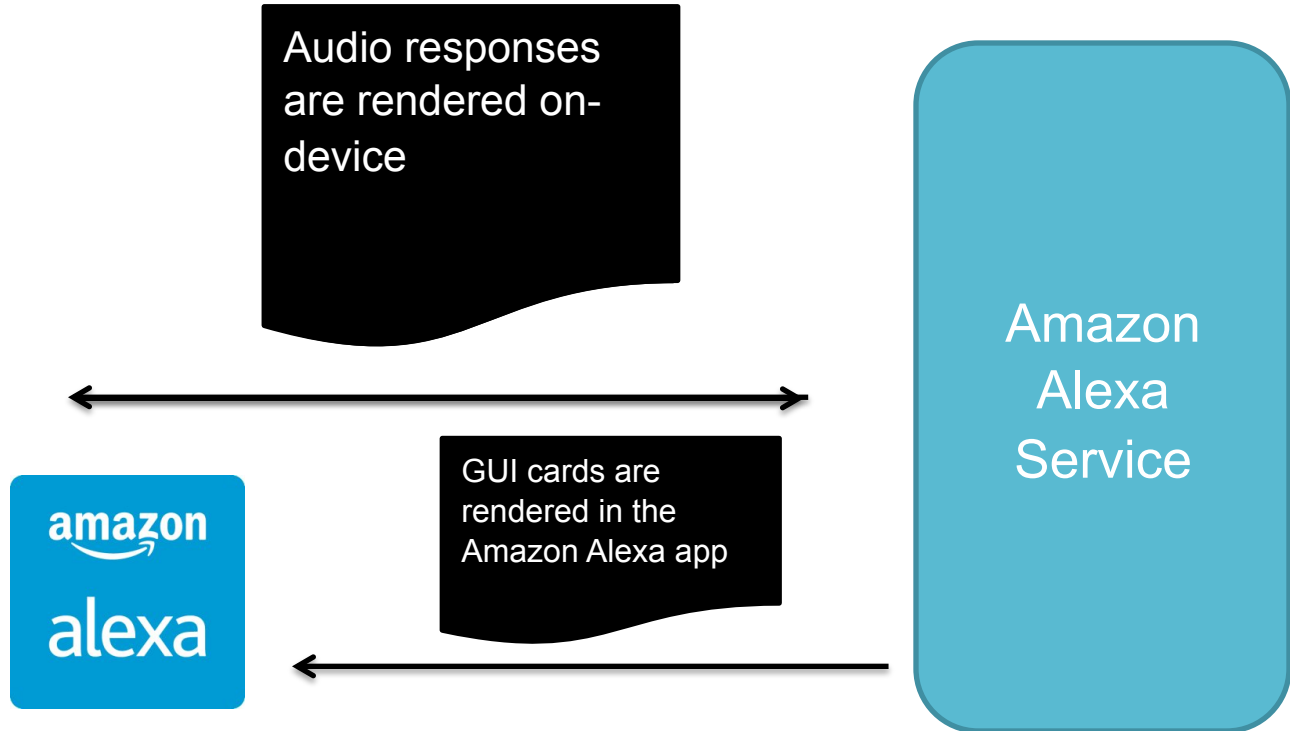
# ALEXA SKILLS KIT

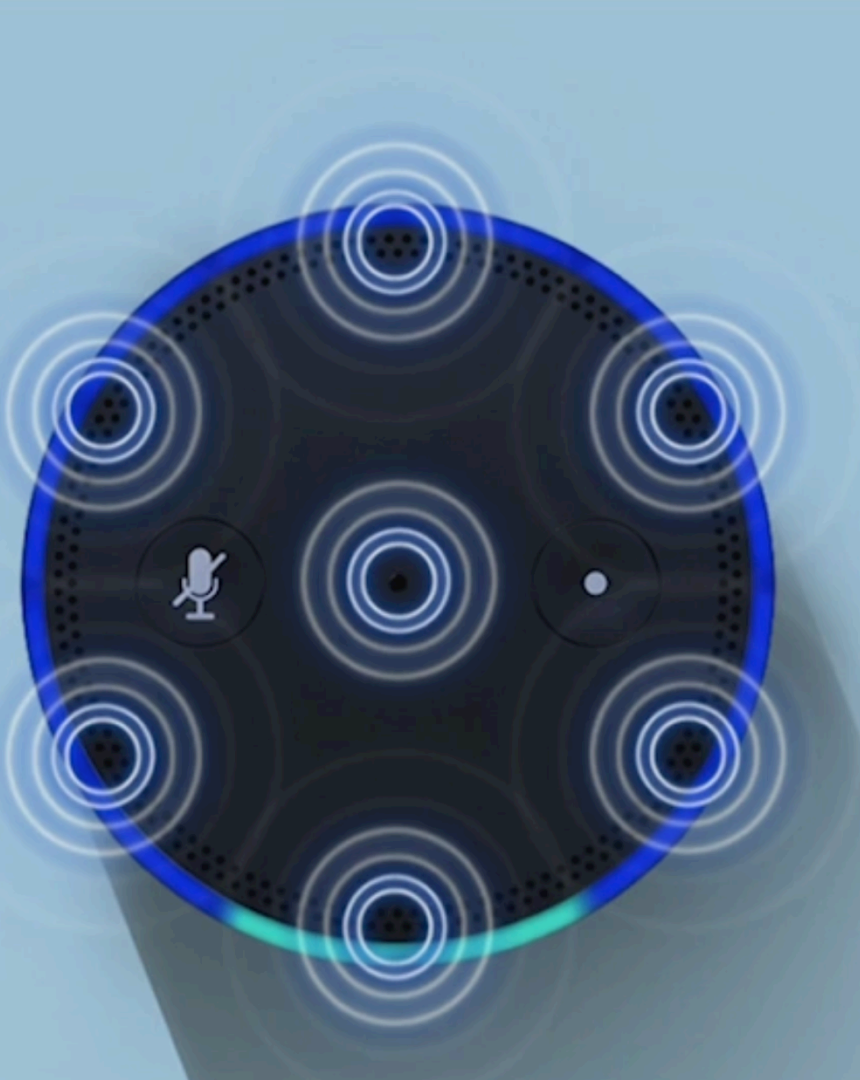
Alexa is the cloud service used by Amazon Echo

The Alexa Skills Kit allows developers to build new Skills (new voice experiences) for Echo

Name	^	Date Modified	Size	Kind
▶  javadoc		Yesterday, 11:01 AM	--	Folder
▶  lib		Yesterday, 11:01 AM	--	Folder
 LICENSE.txt		Apr 15, 2015, 9:09 PM	826 bytes	text
 NOTICE.txt		Apr 15, 2015, 9:09 PM	13 KB	text
 README.txt		Apr 15, 2015, 9:09 PM	1 KB	text
▼  samples		Yesterday, 11:01 AM	--	Folder
 build.xml		Apr 15, 2015, 9:09 PM	917 bytes	XML text
▶  helloworld		Yesterday, 11:01 AM	--	Folder
 Launcher.java		Apr 15, 2015, 9:09 PM	3 KB	Java s...e code
▶  session		Yesterday, 11:01 AM	--	Folder
▶  third-party		Apr 15, 2015, 9:09 PM	--	Folder

# Alexa Architecture





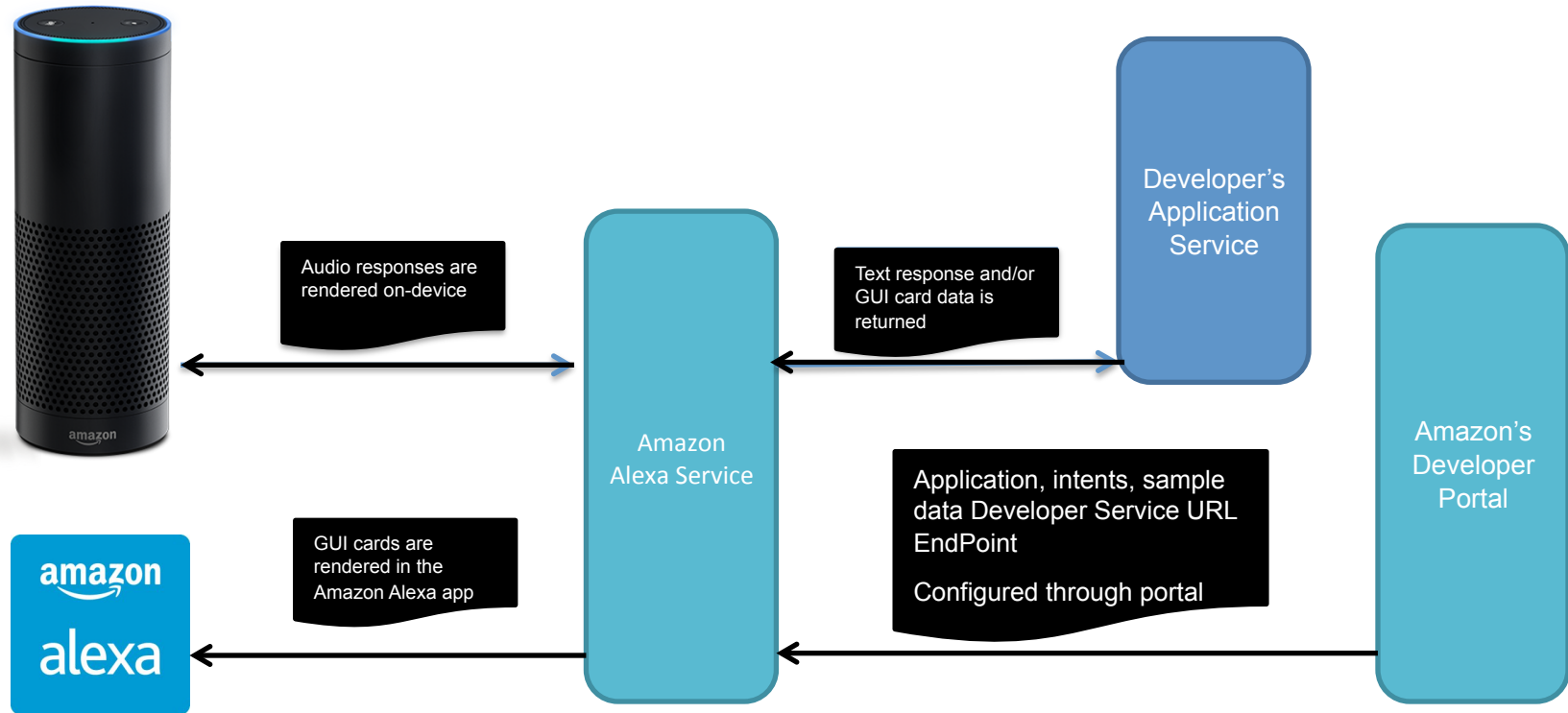
## Creating your own ALEXA SKILLS

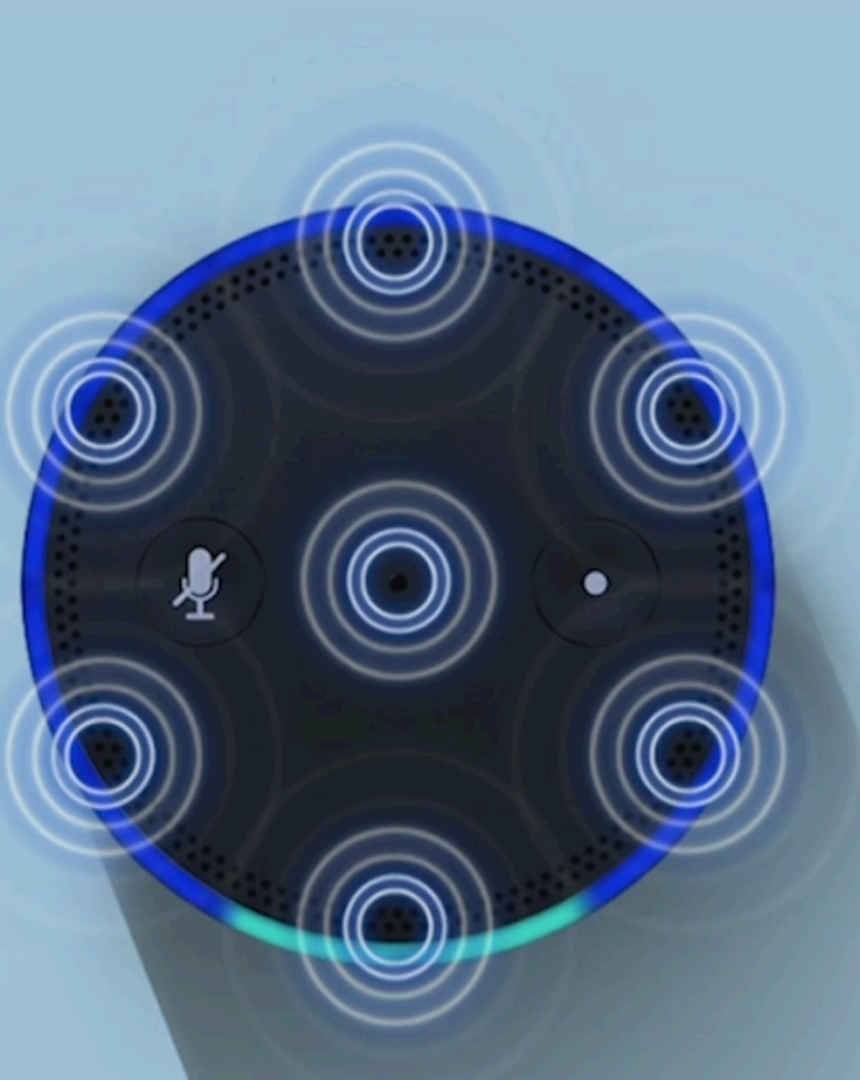
Alexa Skills have two parts:

**Configuration** data in Amazon  
Developer Portal

**Hosted Service** responding to user  
requests

# Alexa Skills Kit Architecture





## Sample Interactions of ALEXA SKILLS

### One and done

“Alexa, start Astrology and get the Pisces horoscope.”

[Horoscope Provided]

### Conversation

“Alexa, start Astrology”

[“What’s your sign?”]

“Pisces”

[Horoscope Provided]

Configuring a new Alexa Skill

# Demo

## Get started with Alexa

Add new voice-enabled capabilities using the Alexa Skills Kit, or add voice-powered experiences to your connected devices with the Alexa Voice Service.



### Alexa Skills Kit

Easily add new skills to Alexa

[Get Started >](#)

### Alexa Voice Service

Bring voice capabilities to your connected device

[Get Started >](#)

DASHBOARD

APPS &amp; SERVICES

REPORTING

SUPPORT

DOCUMENTATION

SETTINGS

My Apps

App Testing Service

Promotions

Security Profiles

Login with Amazon

Cloud Drive

Alexa <sup>New</sup>

GameCircle

A/B Testing

Analytics

PC &amp; Mac Instant Access

Tester Management

Mobile Ads



## Building Alexa Skills with the Alexa Skills Kit

[Add a New Skill](#)

To learn more about building Alexa skills, see [Getting Started with the Alexa Skills Kit](#). To start building an Alexa skill for free in Node.js or Java using AWS Lambda, see [Developing an Alexa Skill as a Lambda Function](#).

We encourage you to visit the [Alexa Developer Forum](#) to collaborate with Alexa team members and fellow Alexa developers.

[Your skills](#)[Skill Metrics](#)

Name	Invocation Name	Version	Modified	Status	Actions
 <b>Appstore Allhands</b>	meeting	1.0	11/16/15	Development ?	<a href="#">Edit</a> <a href="#">Delete</a>
 <b>WhoIsPerson</b>	anyone	1.0	11/15/15	Development ?	<a href="#">Edit</a> <a href="#">Delete</a>
 <b>BerlinColors</b>	berlin colors	1.0	10/23/15	Development ?	<a href="#">Edit</a> <a href="#">Delete</a>
 <b>Daves Colors</b>	daves color	1.00	10/20/15	Development ?	<a href="#">Edit</a> <a href="#">Delete</a>
 <b>myconference</b>	my conference	1.0	10/8/15	Development ?	<a href="#">Edit</a> <a href="#">Delete</a>
 <b>myhackathon</b>	my hackathon		10/8/15	Development ?	<a href="#">Edit</a> <a href="#">Delete</a>
 <b>beanstalkdemo</b>	beanstalk demo	1.0	10/3/15	Development ?	<a href="#">Edit</a> <a href="#">Delete</a>

DASHBOARD

APPS &amp; SERVICES

REPORTING

SUPPORT

DOCUMENTATION

SETTINGS

My Apps

App Testing Service

Promotions

Security Profiles

Login with Amazon

Cloud Drive

Alexa <sup>New</sup>

GameCircle

A/B Testing

Analytics

PC &amp; Mac Instant Access

Tester Management

Mobile Ads

[< Back to the list of skills](#)

## Create a New Alexa Skill

DEVELOPMENT

[Getting started](#)

\*Fields required for certification

Skill Information



Interaction Model



Test



Description



Publishing Information



### Name \*

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

Webinar Colors

### 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](#)

webinar colors

### Version

The serial number of the skill e.g. 1.0, 1.1

1.0

### 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) 

Save

Next



## Building an Alexa Skill HOSTED SERVICE

- Adheres to ASK service interface
- Uses HTTP over **SSL/TLS** on **port 443**
- Must be **Internet-accessible**
- Presents a **trusted certificate** matching domain name
  - Can use self-signed certificate for development
  - Trusted certificate required for certification

```
{
  "intents": [
    {
      "intent": "GetComputer",
      "slots": [
        {
          "name": "Model",
          "type": "MODELS"
        },
        {
          "name": "Date",
          "type": "AMAZON.DATE"
        }
      ]
    }
  ]
}
```

## Building an Alexa Skill HOSTED SERVICE

- You **define** interactions for your skill through **Intent Schemas**
- Each intent consists of two fields. The **intent field** gives the name of the intent. The slots field lists the slots associated with that intent.
- Slots can be any internal types such as AMAZON.LITERAL, AMAZON.NUMBER or they can be ones you create.

# Implementing Custom Slot Types

- **AMAZON.DATE** – converts words that indicate dates (“today”, “tomorrow”, or “july”) into a date format (such as “2015-07-00T9”).
- **AMAZON.DURATION** – converts words that indicate durations (“five minutes”) into a numeric duration (“PT5M”).
- **AMAZON.FOUR\_DIGIT\_NUMBER** - Provides recognition for four-digit numbers, such as years.
- **AMAZON.NUMBER** – converts numeric words (“five”) into digits (such as “5”).
- **AMAZON.TIME** – converts words that indicate time (“four in the morning”, “two p m”) into a time value (“04:00”, “14:00”).

# Implementing Custom Slot Types

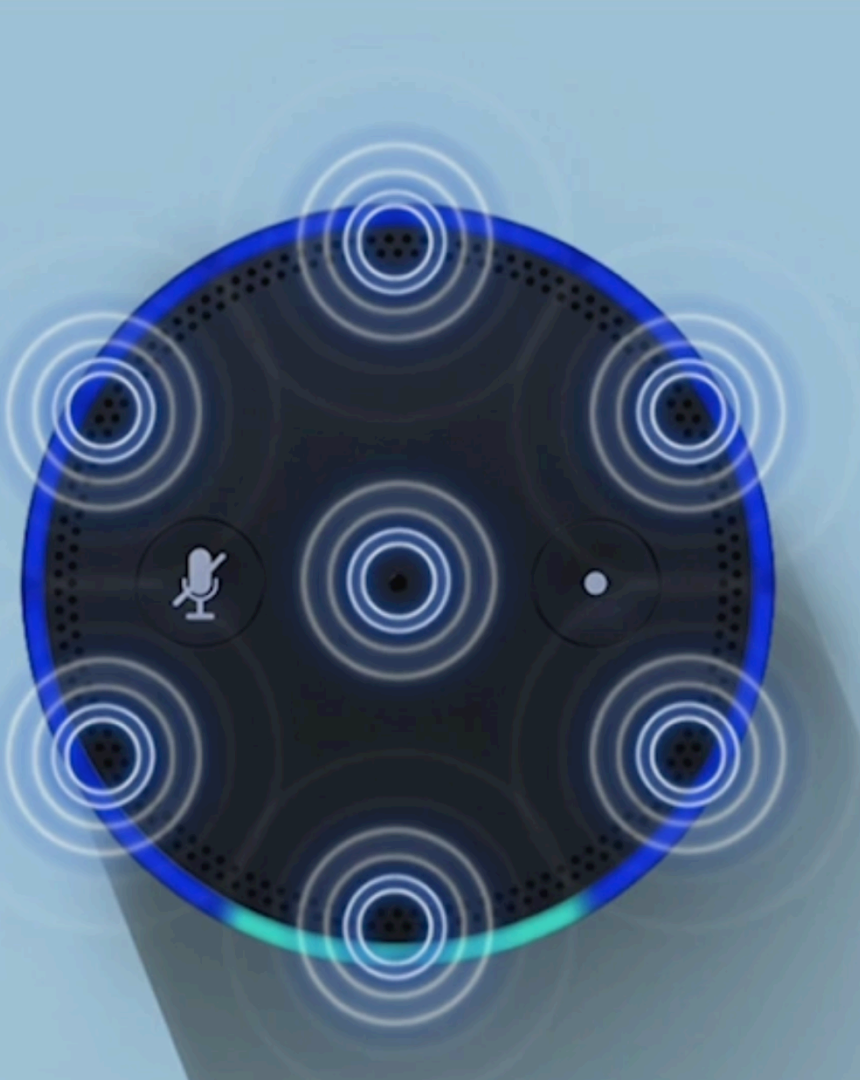
- **AMAZON.US\_CITY** - provides recognition for major cities in the United States.
- **AMAZON.US\_STATE** - provides recognition for US states, territories, and the District of Columbia.
- **AMAZON.US\_FIRST\_NAME** - provides recognition for thousands of popular first names, based on census and social security data.
- **AMAZON.LITERAL** – always try to use a custom slot type first. Used for backwards compatibility.

- “what is...”
- “what’s...”
- “tell me...”
- “give...”
- “give me...”
- “get...”
- “get me...”
- “find...”
- “find me...”

## Building an Alexa Skill

### HOSTED SERVICE

- The mappings between intents and the typical utterances that invoke those intents are provided in a tab-separated text document of sample utterances.
- Each possible phrase is assigned to one of the defined intents.
- **GetModelsBy**Year what were the Amiga models in {1997|Date}
- **GetModel** what year was the {model} released



# The Amazon Alexa Service WILL POST EVENTS

## **LaunchRequest**

Maps to **onLaunch()** and occurs when the user launches the app without specifying what they want

## **IntentRequest**

Maps to **onIntent()** and occurs when when the user specifies an intent

## **SessionEndedRequest**

Maps to **OnSessionEnded()** and when the user ends the session

Examining the JSON Requests from

# Alexa Service

## HTTP Header

```
1 POST / HTTP/1.1
2 Content-Type : application/json;charset=UTF-8
3 Host : your.application.endpoint
4 Content-Length :
5 Accept : application/json
6 Accept-Charset : utf-8
7 Signature:
8 SignatureCertChainUrl: https://s3.amazonaws.com/echo.api/echo-api-cert.pem
```

## Request Body Syntax

The request body sent to an Alexa app is in JSON format.

```
1 {
2   "version": "string",
3   "session": {
4     "new": boolean,
5     "sessionId": "string",
6     "application": {
7       "applicationId": "string"
8     },
9     "attributes": {
10      "string": object
11    },
12    "user": {
13      "userId": "string"
14    }
15  },
16  "request": object
17 }
```

```
{
  "type": "IntentRequest",
  "requestId": "string",
  "timestamp": "string",
  "intent": {
    "name": "string",
    "slots": {
      "string": {
        "name": "string",
        "value": "string"
      }
    }
  }
}
```

## Handling Amazon Alexa Service REQUESTS

- You will need to handle **POST** requests to your service over port 443 and parse the JSON
- You need to check the **session variable** to see if the user started a new session or if request is from existing one
- Requests always Include a **type**, **requestId** and **timestamp**
- requestId maps directly to **LaunchRequest**, **IntentRequest** and **SessionEndedRequest**

```
{
  "intents": [
    {
      "intent": "GetFirstEventIntent",
      "slots": [
        {
          "name": "day",
          "type": "AMAZON.DATE"
        }
      ]
    },
    {
      "intent": "GetNextEventIntent"
    },
    {
      "intent": "AMAZON.HelpIntent"
    },
    {
      "intent": "AMAZON.StopIntent"
    }
  ]
}
```

## Increasing Accuracy with Built-In Intents

- Called in a natural, spontaneous way by users across all skills and Alexa core functionality
- These intents provide a way for you to return help or response whenever these Intents are killed.
- CancelIntent, HelpIntent, YesIntent, NoIntent, RepeatIntent, StartOverIntent, StopIntent

## Custom Slot Types

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

For more information, see [Defining the Voice Interface for an Alexa skill](#).

Example: TOPPINGS - cheese | onions | ham

### Adding slot type

Enter Type \*

MODELS

Enter Values \*

Values must be line-separated

```
1 A500
2 A1000
3 A2000
4 A2000HD
5 A2500
6 A3000
7 A600
8 A1200
9 A4000|
```

## Increasing Accuracy with CUSTOM SLOTS

- Created inside Interaction Model page once in the Developer Portal
- Greatly reduces the number of sample utterances required
- Can define as many as you need with values line separated
- Can be combined with existing AMAZON internal types

# Changing Alexa's inflection with SSML

- Alexa automatically handles normal punctuation, such as pausing after a period, or speaking a sentence ending in a question mark as a question.
- Speech Synthesis Markup Language (SSML) is a markup language that provides a standard way to mark up text for the generation of synthetic speech.
- Tags supported include: speak, p, s, break, say-as, phoneme, w and **audio**.

```
1 <say-as>
2   Here is a number read as a cardinal number:
3   <say-as interpret-as="cardinal">12345</say-as>.
4   Here is the same number with each digit spoken separately:
5   <say-as interpret-as="digit">12345</say-as>.
6   Here is a word spelled out: <say-as type="spell-out">hello</say-as>
7 </say-as>
```

### Account Linking or Creation \*

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

☒ Yes ☐ No

#### Authorization URL \*

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

#### Client Id \*

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

#### 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


HTTP's redirection endpoint uri you want to direct to after completing the authorization interaction with user.

<https://pitangui.amazon.com/sp?vendorid=M295423490KQH9>

## Existing Customer with ACCOUNT LINKING

- Allow your customers to link their existing accounts with you, to Alexa.
- Customers are prompted to log in to your site using their normal credentials with webview url you provide.
- You authenticate the customer and generate an access token that uniquely identifies the customer and link the accounts.

 This Alexa skill is ready for testing

**Enable** ☐ This skill is enabled for testing on this account. 

Once you have completed testing on your device, please complete the Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

**Try this on your Echo: Alexa ask developer day**

---

### Service Simulator

Use Service Simulator to test your lambda function.

**Text** **Json**

**Enter Utterance \***  
when is dave speaking

**Ask Amazon Developer Day** **Reset**

**Lambda Request**

```
{
  "session": {
    "sessionId": "SessionId.b2b555fd-2aa9-44b7-abbe-826d63ddcb01",
    "application": {
      "applicationId": "amzn1.echo-sdk-ams.app.ec9314c5-7e9a-45c2-8318-4b3fe492b8d7"
    },
    "attributes": null,
    "user": {
      "userId": "amzn1.account.AGSRDOGKYM4N2ZYD7GKWH3XAQ5A"
    },
    "new": true
  },
  "request": {
    "type": "IntentRequest",
    "requestId": "EdwRequestId.70ef270c-c860-4f5b-a72a-f22a05a084a2",
    "timestamp": 1441396052972,
    "intent": {
```

**Lambda Response**

```
{
  "version": "1.0",
  "response": {
    "outputSpeech": {
      "type": "plain text",
      "text": "dave will be presenting An Overview of the Amazon Devices and Services for Mobile Developers at 10am. He will also be presenting An Introduction to Using Amazon Web Services and the Alexa Skills Kit to Build Voice Driven Experiences at 10:30am"
    },
    "reprompt": {
      "outputSpeech": {
        "type": "plain text",
        "text": null
      }
    },
    "shouldEndSession": true
  },
  "sessionAttributes": {}
}
```

## Testing Your Skill

# SERVICE SIMULATOR

- Enabled once a Skill has been configured in the Developer Portal
- Use spoken utterances to generate ad hoc results
- Use JSON to verify requests
- Combine with AWS Lambda Unit Tests to verify both client and service side Alexa end points

## Voice Simulator

Hear how Alexa will speak a response entered in plain text or SSML. [Learn more about supported SSML tags.](#)

For example: Here is a word spelled out: <say-as interpret-as="spell-out">hello</say-as>.

hi dave<break time="2s"/>good morning to you.



Listen



## Service Simulator

Use Service Simulator to test your lambda function.

Text

Json

Enter Utterance \*

welcome dave



Ask Appstore Allhands

Reset

### Lambda Request

```
1 {
2   "sessionId": "sessionId.jd/a3a1U-35b1-48fa-",
3   "application": {
4     "applicationId": "amzn1.echo-sdk-ams.app."
5   },
6   "user": {
7     "userId": "amzn1.account.AGSRDOGKYM4N2ZYZ"
8   },
9   "new": true
10 },
11 "request": {
12   "type": "IntentRequest",
13   "requestId": "EdwRequestId.33775759-2d79-42",
14   "timestamp": "1448467359555",
15   "intent": {
16     "name": "IntroSpeakerIntent",
17     "slots": {
18       "Speaker": {
19         "name": "Speaker",
20         "value": "Dave"
21       }
22     }
23   }
24 }
25 }
26 }
```

### Lambda Response

```
1 {
2   "version": "1.0",
3   "response": {
4     "outputSpeech": {
5       "type": "SSML",
6       "ssml": "<speak>Please welcome Dave!</speak>"
7     },
8     "card": {
9       "type": "Simple",
10      "content": "SessionSpeechlet - <speak>Ple",
11      "title": "SessionSpeechlet - IntroSpeaker"
12    },
13    "reprompt": {
14      "outputSpeech": {
15        "type": "PlainText",
16        "text": null
17      }
18    },
19    "shouldEndSession": true
20  }
21 }
```

Listen



# Testing Your Skill VOICE SIMULATOR

- The new Voice Simulator tool will let you create SSML on the fly and then have Alexa respond.
- The Voice Simulator does not interact with the service for your skill, so you can use it to hear text-to-speech conversion before you have completed the minimum configuration for your skill.



# AWS Lambda

---

Run code without thinking about servers.  
Pay for only the compute time you consume.

[Get started with AWS Lambda](#)

Creating a new Alexa skill with AWS Lambda

# Demo



**<http://developer.amazon.com/ask>**

**<http://developer.amazon.com/blog>**