

Alexa, Ask Mura...

ALEXA SKILLS WITH MURA

blueriver

mura
2018
con

Paul Denato

WEB DEVELOPER ~ BLUERIVER

This Session

ALEXA, ASK MURA...



Developer Portal

JUST THE BASICS

Alexa Interactions

INTENTS, UTTERANCES AND SLOTS

Lambda Function

CONNECTING TO THE API

Demo

DEMO SKILLS WITH ALEXA

AWS

DEVELOPER PORTAL



The screenshot shows the Amazon Developer Services and Technologies landing page. At the top right are links for "Developer Console", "Sign In", a help icon, and a search icon. The main title "Amazon Developer Services and Technologies" is centered above five service cards:

- amazon alexa**: Alexa. Build natural voice experiences that offer customers a more intuitive way to interact with technology.
- amazon appstore**: Amazon Appstore. Develop Android apps and games for Amazon Fire TV, Fire Tablet, and mobile platforms.
- aws**: Amazon Web Services. Reliable, scalable, and inexpensive cloud computing services.
- Desktop Apps & Games**: Desktop Apps & Games. Develop apps and games for Windows and Mac.
- amazon dash services**: Dash Services. Build Amazon reordering experiences into your devices.

<https://developer.amazon.com>

Alexa

SKILLS CONSOLE

The screenshot shows the Alexa Skills Kit Developer Console interface. The top navigation bar includes tabs for Hey Mura, Build, Test, Launch, and Measure. The Build tab is active. A sidebar on the left lists sections: English (U.S.), CUSTOM, Interaction Model, Invocation (Intents: 4, Add), Slot Types (2, Add), and Resources (Documentation, Sample Alexa Projects, Weekly Office Hours, Alexa Developer Forums). The main content area features a "How to get started" section with a "Alexa Skills Kit Developer Console: Build" link and a "Skill builder checklist" on the right. The checklist consists of four required steps, each with a green checkmark:

- 1. Invocation Name >** Enter an invocation name for your skill
- 2. Intents, Samples, and Slots >** Add at least one intent and one sample utterance
- 3. Build Model >** Successfully build your interaction model
- 4. Endpoint >** Set a web service endpoint to handle skill requests

At the bottom, a footer bar includes links for Terms, Alexa Developer Blog, and Alexa Skills Kit.

Alexa

INVOCATION

The screenshot shows the Alexa Skills Kit (ASK) console interface. The top navigation bar includes links for Hey Mura, Build, Test, Launch, Measure, Your Alexa Consoles, PD, ?, and a search icon. Below the navigation is a toolbar with Save Model and Build Model buttons.

The main content area is titled "Invocation". It explains what an invocation name is and provides an example of a user interaction:

User: Alexa, ask daily horoscopes for the horoscope for Gemini

The "Skill Invocation Name" field contains the value "mura".

Invocation name requirements

Your invocation name should be two or more words, and can contain only lower-case alphabetic characters, spaces between words, possessive apostrophes (for example, "sam's science trivia"), or periods used in abbreviations (for example, "a. b. c."). Other characters like numbers must be spelled out. For example, "twenty one".

Invocation names cannot contain any of the Alexa skill launch phrases such as "launch", "ask", "tell", "load", "begin", and "enable". Wake words including "Alexa", "Amazon", "Echo", "Computer", or the words "skill" or "app" are not allowed. [Learn more](#) about invocation names for custom skills.

The left sidebar lists the configuration sections: English (U.S.), CUSTOM, Interaction Model, and Invocation. Under Invocation, there are sections for Intents (4), Slot Types (2), and Built-In Intents (3). The Intents section shows four items: AddIntent (highlighted), IdeaName, IdeaContent, IdeaRating, and three built-in intents: AMAZON.CancelIntent, AMAZON.HelpIntent, and AMAZON.StopIntent.

Intents, utterances and slots

ALEXA INTERACTIONS

Intent: Actions that users can do with your skill. They represent the core functionality for your skill.

YOUR SKILL CAN HAVE MANY INTENTS

INTENTS

ADD, GET, DELETE



Add An Idea

Get An Idea

Delete An Idea

Get All Ideas

Get Ideas From Yesterday

Utterances: A set of words or likely spoken phrases mapped to the intents.

THESE WILL TELL ALEXA WHICH INTENT TO CALL

UTTERANCES

USER SPOKEN PHRASES



`addAnIdeaIntent`

Add an idea

I have an Idea

I just had a great idea

I have another idea

Slots: A representative list of possible arguments for an intent.

LIKE METHODS. RIGHT?

Alexa

CUSTOM SLOT

The screenshot shows the Alexa Skills Kit (ASK) developer console interface. On the left, a sidebar lists several sections: Intents (4), Slot Types (2), JSON Editor, Interfaces, Endpoint, ACCOUNT LINKING, and PERMISSIONS. The Intents section is expanded, showing four intents: AddIdeaIntent (selected), IdeaContent, IdeaRating, and three built-in intents: AMAZON.CancelIntent, AMAZON.HelpIntent, and AMAZON.StopIntent. The Slot Types section shows two types: IdeaRating and AMAZON.Service.

The main panel displays the configuration for the 'IdeaName' slot under the 'AddIdeaIntent' intent. A note at the top says: "To use these prompts and utterances in your skill, return the `Dialog.Delegate` directive in your skill's response. Learn more about using Dialog directives." Below this is a "Slot Filling" section. A toggle switch is set to "On" for "Is this slot required to fulfill the intent?". The "Alexa speech prompts" section contains a text input field with the placeholder "What will Alexa say to prompt the user to fill this slot?". The "User utterances" section contains a text input field with the placeholder "What might a user say in response to the above prompt(s?)". A template placeholder "{IdeaName}" is shown in the bottom input field.

Alexa ask Mura to add an idea.

What is the name of the idea?

Bacon Pancakes

Lambda

NODEJS FUNCTION TO MAKE API CALLS

Lambda

ADD IDEA CONTENT

```
// IdeaContent
if (!slots.IdeaContent.value) {
  const slotToElicit = 'IdeaContent';
  const speechOutput = 'So what is your idea?';
  const repromptSpeech = 'Please give me the content for this idea.';
  return this.emit(':elicitSlot', slotToElicit, speechOutput, repromptSpeech);
}

else if (slots.IdeaContent.confirmationStatus !== 'CONFIRMED') {

  if (slots.IdeaContent.confirmationStatus !== 'DENIED') {
    // slot status: unconfirmed
    const slotToConfirm = 'IdeaContent';
    const speechOutput = `The idea is ${slots.IdeaContent.value}, correct?`;
    const repromptSpeech = speechOutput;
    return this.emit(':confirmSlot', slotToConfirm, speechOutput, repromptSpeech);
  }

  // slot status: denied -> reprompt for slot data
  const slotToElicit = 'IdeaContent';
  const speechOutput = 'So what is your idea?';
  const repromptSpeech = 'Please give me the content for this idea.';
  return this.emit(':elicitSlot', slotToElicit, speechOutput, repromptSpeech);
}
```

Lambda

TALK TO MURA

```
//Slot Values
const muraParams = {
  'message': {
    'title': iname,
    'remoteid': userId,
    'contentbody': icontent,
    'irating': irating
  }
};

// method
var servicenamesetter = '/index.cfm/_api/rest/v1/alexa/askmuraservice/addAnIdea/';
// append path with method
options.path = servicenamesetter;
// append body to post call
var myNewIdea = JSON.stringify(muraParams);

options.headers = {
  'Cache-Control': "no-cache",
  'Content-Type': "application/json",
  'Authorization': "Basic " + apiAuth,
  'Content-Length': Buffer.byteLength(myNewIdea)
};

sendMuraData(options,myNewIdea).then((json) => {
  var md = JSON.parse( json );
  var isMuraIdeaAdded = "";
  isMuraIdeaAdded=md.data;

  //Anything other than true is considered an error.
  if (isMuraIdeaAdded.added) {
    const successMessage = `Idea ${iname} added, and may I say<brak strength="medium" />It's wonderful!`;
    this.emit(':tellWithCard', successMessage , iname, icontent);
  } else {
    const errorMsg = `You already had the ${iname} idea. Great Idea by the way.`;
    this.emit(':tellWithCard', errorMsg, iname, icontent);
  }
})
```

Demo

LET'S TEST THIS OUT!



Questions?

NO, PERFECT!

Thank You!