



Alexa,  
let's build a voice-Bot

# How to build a voicebot and get money for it :-)

1. Why create a voice Bot?
2. What do we need?

# Who I am

Lectures, consultancy & workshops for 20+ years

Google Developer Launchpad Mentor

SW development with small team

...and last four years chatBots & Voice Assistants

# Why create voiceBot?



# Speaking Age is back

- Storytelling from the caveman, Gutenberg via emoji..back to voice
- How to learn when you cannot read (age <5 ... or age to 15?)
- How to read and write when you cannot see (60? or 40+)
- When you are in your car
- How to control smart home when you are disabled



# Listen & Talk



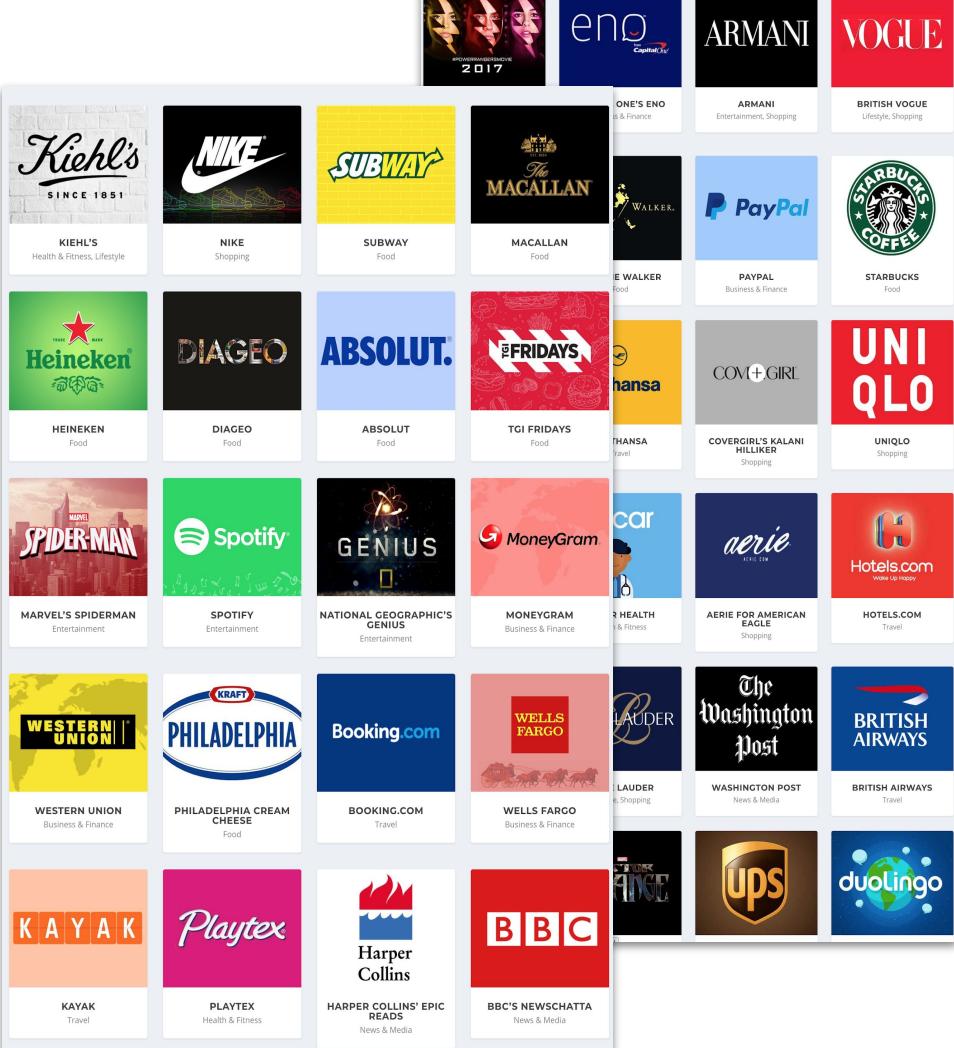
# Nice “fight” Amazon vs Google (vs others)

- Siri: Out of a clear blue sky
- Echo: 2+ years takes to create the Echo working in an “open room”
- Unbelievable prices drop down to 30USD, alza.cz 1 290 CZK
- Many variations (car, watch, phone, show, spot, speaker, TV, headphones  
...Echo LOOK)



# People love it

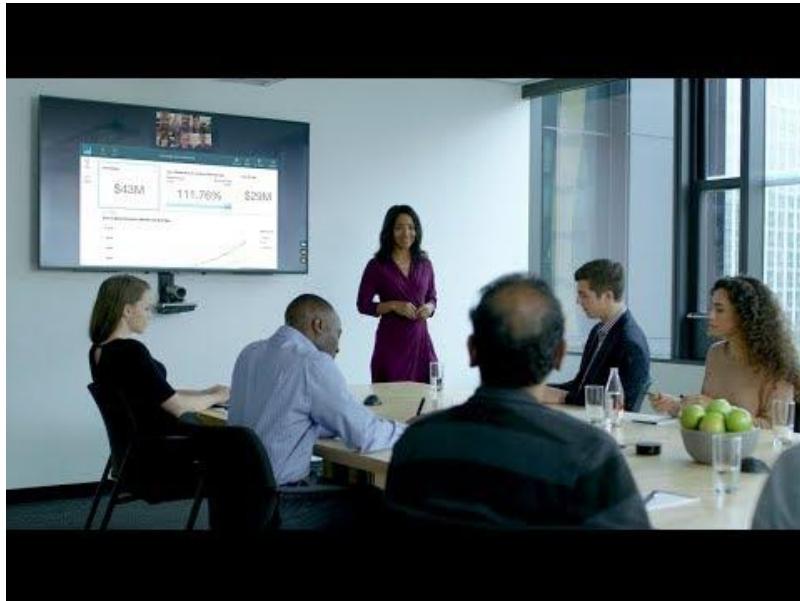
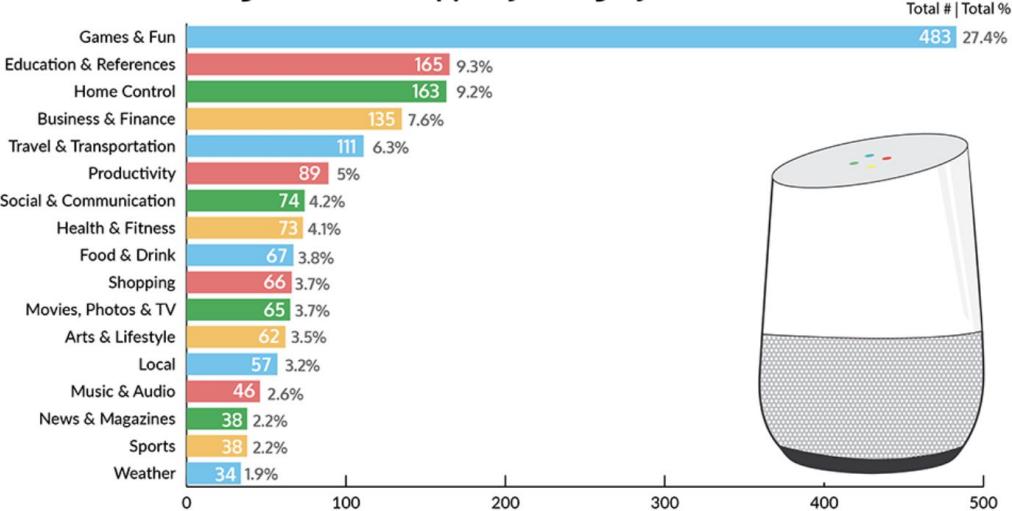
- Companies paying for chat & voice bots
- Big brands and many use-cases



# From Fun to Business

- Started at home, now in business
- Alexa for Business Services

Google Assistant Apps by Category - December 2017



# Let's build a voice-Bot



# What do we need?

1. The Idea
2. Preparation “intellectual” part
  - Strategy, Brand identity, Persona, Conversation design, Targeted devices, Errors handling, Testing ... Marketing :-/
3. Coding “VIP” part
  - Registration AWS, Alexa Skills Services & Interface
4. Money talks “happy” part

# The Idea

Alexa, ask Greeting BOT

Hi Martin, who you want to say hello?

Say hello to Petr

Hey Peter, nice to meet you!



# Preparation “intellectual” part (1/2)

**Strategy:** Why I do it and what is my goals. How will measure success? What will be the next steps?

**Brand identity & Persona:** Image that a voice bot is a real person. What will be his "properties," mentality, language & jargon... Google voices, Alexa still the same voice, and "recorded voices."

**Users:** who is your user? How old? Does he speak "your way"? Where they use it?

**Conversation design:** Start with a Happy path, write dialogues, read them and record the conversation. Add new branches and do it again.

# Preparation “intellectual” part (2/2)

**Targeted devices:** car, watch, phone, show, spot, look, speaker, TV, headphones  
... where can I talk only and where can I interact.

**Error handling:** all possible and impossible conversation branches,  
Problematic answer: “*I don't understand*”

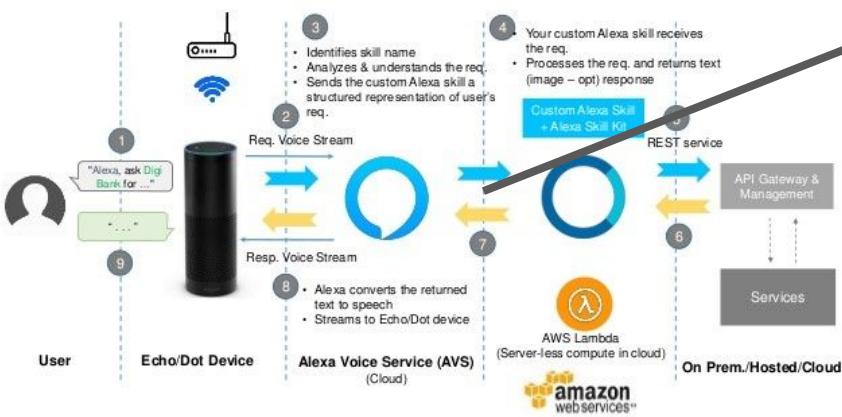
**Testing:** Automation, real people from the target group.  
CPU is running but the bot is not responding?

**Marketing:** How will I invite users, Where will I get sounds & voices? Who will  
answer the reviews? How can I do a promotion?

# Coding “VIP” part

# Architecture

Alexa Custom Skill - Reference Architecture



Skill

Skill Interface  
Skill Service

- JSON
- Lambda functions (JavaScript, Python, ...)
- AWS Cloud9 IDE
- External https server (Hosting, RPi)

# Greeting Bot

Alexa, ask Greeting BOT

Hi Martin!



# Greeting Bot

Invocation name for Skill



Alexa, ask **Greeting Bot**



# Skill Interface

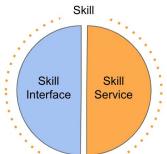
[developer.amazon.com/alexza/console](https://developer.amazon.com/alexza/console)

## New Skill

Skill Name: Greeting Bot

Choose a template: Start from scratch

Invocation name: greeting bot



The screenshot shows the 'Create a new skill' page in the Alexa developer console. The skill name is set to 'Greeting Bot' and the default language is 'English (US)'. A callout arrow points to the 'Custom' model under 'Choose a model to add to your skill'. Other models shown include 'Flash Briefing', 'Smart Home', and 'Video'.

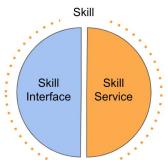
The screenshot shows the 'Choose a template' page in the Alexa developer console. The 'Start from scratch' template is selected, indicated by a blue triangle icon. Other templates shown include 'Fact Skill', 'Quiz Game Skill', and 'High-Low Game Skill'.

The screenshot shows the 'Interaction Model' configuration page in the Alexa developer console. Under the 'Invocation' section, the invocation name is set to 'greeting bot'. A callout arrow points to the 'name' field in the 'Invocation' section. The 'Skill Invocation Name' field also contains 'greeting bot'. Below the invocation setup, there is a section titled 'Invocation name requirements' with some explanatory text.

# Skill Interface

## Build Model

- Successfully build your interaction model



The screenshot shows the Alexa Skills Kit Developer Console interface. The top navigation bar includes tabs for Your Skills, Greeting Bot, Build, Test, Distribution, Certification, and Analytics. The Build tab is active. On the left, a sidebar titled "CUSTOM" contains sections for "Interaction Model" (Invocation, Intents, Slot Types), "Resources" (Make Money, Documentation, Sample Alexa Projects, Weekly Office Hours), and "Interfaces". The main content area displays a "How to get started" guide for the "Alexa Skills Kit Developer Console: Build". This guide includes a simulator window showing the Amazon Alexa logo and the text "Developer Console: Build", a "Skill builder checklist" with four items (Invocation Name, Intent Samples, Build Model, Endpoint), and links for In-Skill Purchasing (ISP) and weekly office hours. The checklist items 1, 2, and 3 have green checkmarks, while item 4 is marked as optional.

This screenshot is identical to the one above, showing the Alexa Skills Kit Developer Console interface in the Build tab. It displays the "How to get started" guide, the "Skill builder checklist" with four items, and the "Resources" section. All checklist items (1, 2, 3, 4) have green checkmarks indicating they are completed.

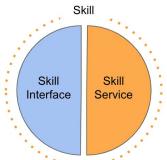
# Skill Interface

## Endpoint

- Set a web service endpoint to handle skill requests

The screenshot shows the 'Interaction Model' section of the ASK console. Under 'Invocation', there are 'Intents (6)' and 'Slot Types (1)'. Under 'Interfaces', the 'Endpoint' tab is selected. On the right, the 'Endpoint' configuration page is displayed. It includes a note about receiving POST requests and a JSON template. The 'Service Endpoint Type' section shows 'AWS Lambda ARN (Recommended)' selected. The 'Your Skill ID' is listed as 'amzn1.ask.skill.37c10ed0-522c-4c7b-bd8f-bb0525'. Below this, 'Default Region' is set to 'Reserved'. Regions listed include North America (Optional), Europe and India (Optional), and Far East (Optional), each with a corresponding ARN placeholder.

...continue on Skill Services: Lambda Functions



# Skill Service

[console.aws.amazon.com/lambda](https://console.aws.amazon.com/lambda)

## New function

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Author from scratch' tab is selected. The form fields include:

- Name: myFunctionName
- Runtime: Node.js 6.10
- Role: Choose an existing role
- Existing role: You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

At the bottom right are 'Cancel' and 'Create function' buttons.

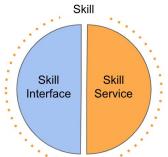
## Author from scratch or Blueprints

alex-skills-kit-sdk-facts  
skill  
(nodejs6.10-alexa)

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Blueprints' tab is selected. The interface includes:

- Blueprints info: A search bar and a page navigation bar (1-11).
- A grid of blueprint cards:

  - kinesis-firehose-syslog-t-o-jon
  - logimonitor-send-cloud-watch-events
  - splunk-elb-application-a-process-logs-processor
  - alexa-skills-kit-sdk-factsk-ill
  - batch-get-job-python27
  - kinesis-firehose-apache-log-to-json-python
  - s3-get-object-python
  - config-rule-change-triggered
  - lex-book-trip-python
  - dynamodb-process-stream



# Skill Service

Name: greetingFunction

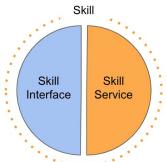
Role: Choose an existing role

Existing role: lambda\_basic\_execution

This screenshot shows the AWS Lambda function configuration interface. At the top, a message says "This function contains external libraries." Below it, the "Basic information" section has a "Name" field set to "greetingFunction". Under the "Role" section, it says "Choose an existing role" and "lambda\_basic\_execution" is selected. In the "Lambda function code" section, it says "Code is pre-configured by the chosen blueprint. You can configure it after you create the function." Below that, the "Runtime" is listed as "Node.js 6.10". The code editor contains the following Node.js code:

```
1 /* eslint-disable func-names */
2 /* eslint quote-props: ['error', 'consistent'] */
3 /**
4 * This sample demonstrates a simple skill built with the Amazon Alexa Skills
5 * nodejs skill development kit.
6 * This sample supports multiple languages. See LS_en-US_de-DE
7 */
```

Create a function



# Hook up

## Connect Skill Interface & Service

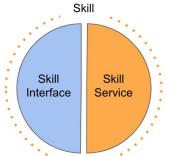
### Add Alexa Skill

- add Your Skill ID from Skill Interface (Endpoint)

### Add Service Endpoint

- AWS Lambda ARN from Skill Services (ARN)

Default Region

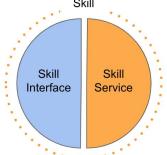


The screenshot shows the AWS Lambda console and the Alexa developer console side-by-side, connected by orange arrows.

**AWS Lambda Console:** The top part shows the 'greetingFunction' configuration. It has an 'Add triggers' section with a key icon, an 'Actions' dropdown set to 'Select a test event...', and a 'Test' button. The ARN 'arn:aws:lambda:us-west-2:05078455:function:greetingFunction' is displayed at the top right. A yellow smiley face icon is placed over the ARN.

**Alexa Developer Console:** The bottom part shows the 'Your Skills' section. Under 'Interaction Model', it lists 'Invocation' and 'Intents (6)'. Under 'Service Endpoint Type', 'AWS Lambda ARN' is selected, and the 'Your Skill ID' field contains 'amzn1.task.skill.37c1dce0-522c-4c7b-bd8f-8b05237c'. A yellow smiley face icon is placed over the skill ID. Below this, the 'Default Region' dropdown is set to 'us-west-2'. The ARN 'arn:aws:lambda:us-west-2:05078455:function:greetingFunction' is shown again, with a yellow smiley face icon over it. A 'Clipboard' icon is also present next to the ARN. A yellow smiley face icon is also placed over the 'Skill Manifest Saved Successfully' message at the bottom.

# Skill Interface: test



The screenshot shows three windows of the Alexa Developer Console:

- Top Window:** Shows the 'Test' tab selected. It displays a message: "Test is disabled for this skill. When test is enabled, you can interact with the development version of your skill in the Alexa simulator and on all devices linked to your Alexa developer account." Below this is a large input field labeled "Type or click and hold the mic".
- Middle Window:** Shows the 'Test' tab selected. It displays a message: "First, open your skill with your invocation name. Then start testing your dialog." Below this is a large input field labeled "Type or click and hold the mic".
- Bottom Window:** Shows the 'Test' tab selected. It displays a message: "Skill I/O is available only f..." (partially cut off). Below this is a large input field labeled "Type or click and hold the mic".

**Right Side:**

- JSON Input:**

```
1 - {  
2 -   "version": "1.0",  
3 -   "session": {  
4 -     "new": true,  
5 -     "sessionId": "amzn1.echo-api.session.832e9112-5f06-485c-0",  
6 -     "application": {  
7 -       "applicationId": "amzn1.ask.skill.37c1dce0-522c-4c7b-1",  
8 -     },  
9 -     "user": {  
10 -       "userId": "amzn1.ask.account.AFDSYCFLO3TA77VUQVGYMLU",  
11 -     },  
12 -     "context": {  
13 -       "System": {  
14 -         "application": {  
15 -           "applicationId": "amzn1.ask.skill.37c1dce0-522c-4",  
16 -         },  
17 -         "user": {  
18 -           "userId": "amzn1.ask.account.AFDSYCFLO3TA77VUQVGY",  
19 -         },  
20 -         "device": {  
21 -           "deviceId": "amzn1.ask.device.AG3ROTKNFESYGSFW4Y",  
22 -           "supportedInterfaces": {}  
23 -         },  
24 -         "apiEndpoint": "https://api.amazonalexa.com",  
25 -         "apiAccessToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Ni",  
26 -       }  
27 -     },  
28 -     "request": {}  
29 -   }  
30 - }
```
- JSON Output:**

```
1 - {  
2 -   "body": {  
3 -     "version": "1.0",  
4 -     "response": {  
5 -       "outputSpeech": {  
6 -         "type": "SSML",  
7 -         "ssml": "Hi Martin, who you want to say hello?",  
8 -       },  
9 -       "shouldEndSession": false  
10 -     },  
11 -     "sessionAttributes": {},  
12 -     "userAgent": "ask-nodejs/1.0.25 Node/v6.18.3"  
13 -   }  
14 - }
```
- Status Bar:**

Build Successful

If you make any new changes, you will need to rebuild your model for them to take effect.

Go to build

# Greeting Bot

Alexa, ask Greeting BOT

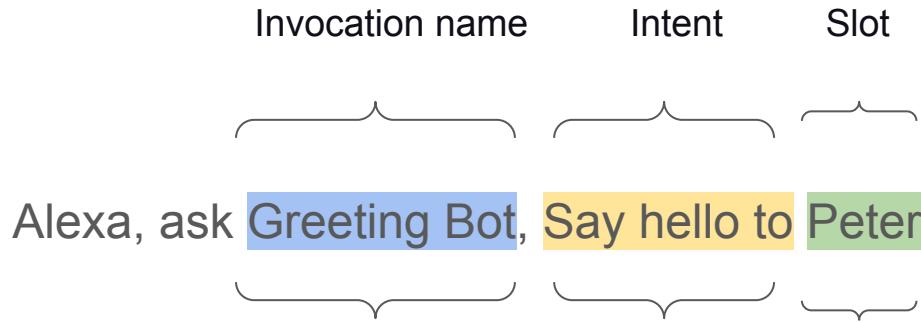
Hi Martin, who you want to say hello?

Say hello to Petr

Hey Peter, nice to meet you!



# Greeting Bot



# Skill Interface: Intent

Add New Intent: SayHalloIntent

The screenshot shows the Alexa developer console interface. In the top navigation bar, the 'Build' tab is selected. On the left sidebar, under the 'CUSTOM' section, the 'Intents' item is expanded, showing 'SayHalloIntent' listed. The main content area is titled 'Add Intent' with the sub-instruction 'An intent represents an action that fulfills a user's spoken request.' Below this, there are two options: 'Create custom intent' (selected) and 'Use an existing intent from Alexa's built-in library'. The 'Create custom intent' section contains a text input field with 'SayHalloIntent' and a blue 'Create custom intent' button. The 'Use an existing intent from Alexa's built-in library' section shows a search bar with 'Search built-ins' and a list of categories: 'Books' (16 built-ins), 'Calendar' (2 built-ins), and 'Cinema Showtimes' (4 built-ins). On the right side of the screen, a green circular icon indicates a 'Build Successful' status, with a note: 'If you make any new changes, you will need to rebuild your model for them to take effect.' At the bottom right, the date and time are shown: 'Tuesday, September 18, 2018, 1:01 PM'.

Sample Utterances (Training phrases)

Say hello to Peter

Peter

to Peter

The screenshot shows the Alexa developer console interface, specifically the 'Intents / SayHalloIntent' page. The top navigation bar has the 'Build' tab selected. The left sidebar shows the 'Intents' section with 'SayHalloIntent' selected. The main content area is titled 'Intents / SayHalloIntent' and contains a 'Sample Utterances (2)' section with three entries: 'to Peter', 'Petr', and 'Say hello to Peter'. Below this is the 'Intent Slots (0)' section, which includes a table with columns for ORDER, NAME, SLOT TYPE, and ACTIONS. There is one row in the table with the value '1' in the ORDER column, 'Create a new slot' in the NAME column, and 'Select a slot type' in the SLOT TYPE column. At the bottom of the screen, a green circular icon indicates a 'Build Successful' status, with a note: 'If you make any new changes, you will need to rebuild your model for them to take effect.' At the very bottom right, the date and time are shown: 'Tuesday, September 18, 2018, 1:01 PM'.

# Skill Interface: Slot

Say hello to Peter

Change “Peter” -> variable name

Say hello to {name}

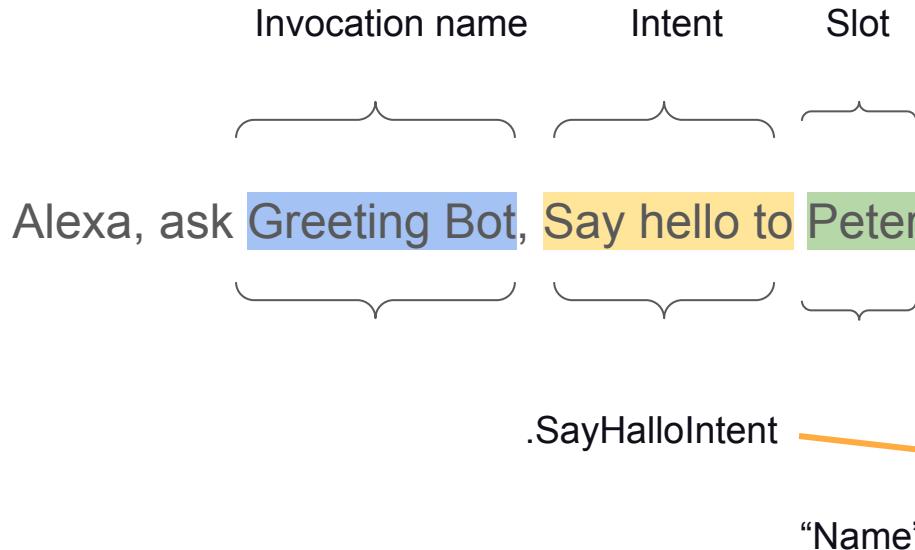
## Slot type

- AMAZON (names, days, numbers...)
- own type (own variables)

The screenshot shows the Alexa Developer Console interface. On the left, the navigation bar includes 'Your Skills', 'Greeting Bot', 'Build', 'Test', 'Distribution', 'Certification', and 'Analytics'. The main panel is titled 'Intents / SayHelloIntent'. It shows sample utterances like 'to Peter' and 'Petr'. A modal window titled 'Select an Existing Slot' lists 'No existing slots'. Below it, the 'Intent Slot' section shows a slot named 'Peter' with an 'Add' button. The 'Intent Confirmation' section has a toggle switch labeled 'Does this intent require confirmation?'. The bottom right corner of the screen displays the text '1 – 3 of 3'.

This screenshot shows the same Alexa Developer Console interface, but with a different configuration. The 'Slot Type' dropdown in the 'Intent Slot' section is set to 'Name'. The 'Intent Confirmation' section now has the toggle switch turned off. The bottom right corner of the screen displays the text '1 – 3 of 5'.

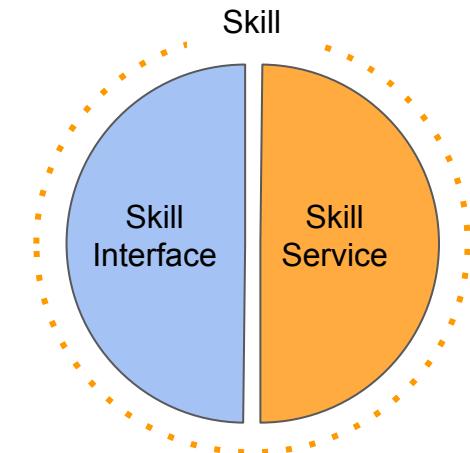
# Skill Interface: JSON Editor (overview)



The screenshot shows the AWS Lambda Interaction Model JSON Editor. The left sidebar lists the "Invocation Model" with sections for "Invocation", "Intents (6)", and "Slot Types (1)". The "Intent" section is expanded, showing entries for "SayHalloIntent" (with a "Name" slot) and other built-in intents like "AMAZON.FallbackIntent". The "Slot Types" section shows "AMAZON.US\_FIRST\_NAME". The right panel is the "JSON Editor" where the interaction model's JSON code is displayed. Two orange arrows point from the "Intent" and "Slot" labels in the diagram above to the corresponding parts in the JSON code. The JSON code is as follows:

```
1 <= {
2   "interactionModel": {
3     "languageModel": {
4       "invocationName": "greeting bot",
5     },
6     "intents": [
7       {
8         "name": "AMAZON.FallbackIntent",
9         "samples": []
10      },
11      {
12        "name": "AMAZON.CancelIntent",
13        "samples": []
14      },
15      {
16        "name": "AMAZON.HelpIntent",
17        "samples": []
18      },
19      {
20        "name": "AMAZON.StopIntent",
21        "samples": []
22      },
23      {
24        "name": "AMAZON.NavigateHomeIntent",
25        "samples": []
26      }
27    ],
28    "name": "SayHalloIntent",
29    "slots": [
30      {
31        "name": "Name",
32        "type": "AMAZON.US_FIRST_NAME"
33      }
34    ],
35    "samples": [
36      "to {Name}",
37      "{Name}",
38      "Say hello to {Name}"
39    ]
40  }
41 }
```

# Skill Interface + Services Lambda Functions



 Save Model

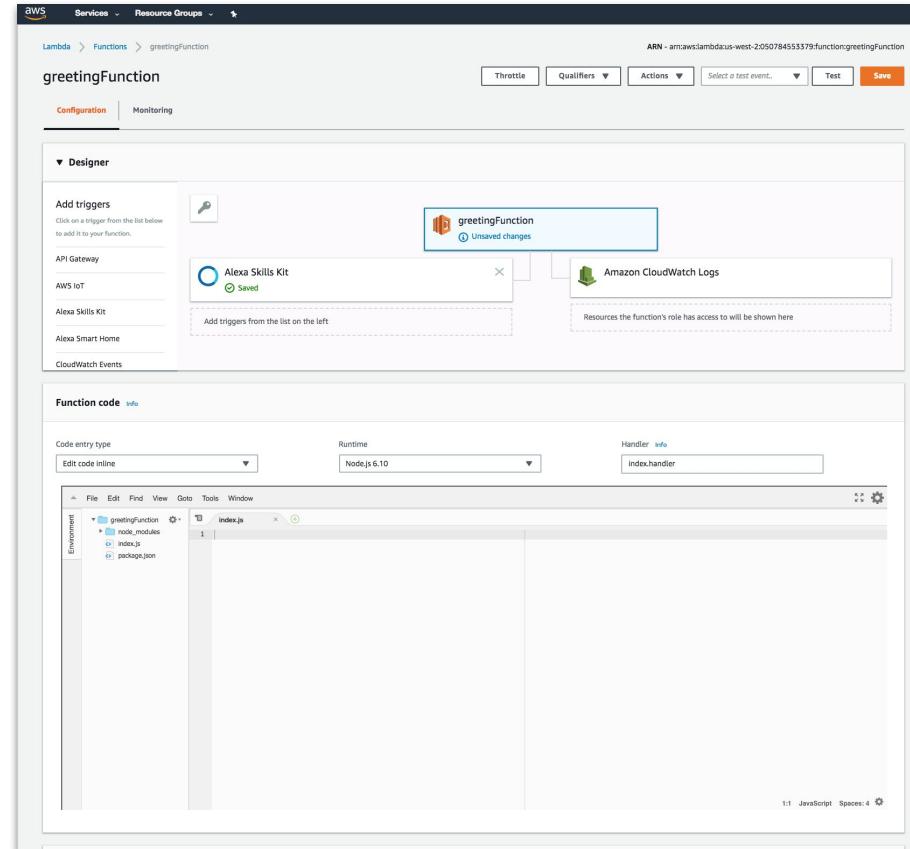
 Build Model

**Save**

# Skill Services

## Online editor (Cloud9)

- Delete the “Blueprint template”
- Write own code



# Skill Services

LaunchRequest - invoked on launch

WelcomeIntent - ask and wait for answer

SayHalloIntent - our great function

Slot (variable):

this.event.request.intent.slots.name.value

Invocation name      Intent      Slot

Alexa, ask Greeting Bot, Say hello to Peter

.SayHalloIntent

"name"

```
index.js x +  
1  'use strict';  
2  const Alexa = require('alexa-sdk'); //import Alexa SDK  
3  
4  //---  
5  //Messages for our Prompt  
6  //---  
7  //  
8  const APP_ID = undefined;  
9  const SKILL_NAME = 'Greeting Bot';  
10 const SKILL_WELCOME = 'Hi Martin, who you want to say hello?';  
11 const HELP_MESSAGE = 'Just say a name, and I will say hello to this person';  
12 const HELP_REPROMPT = 'Say: Say hello to John';  
13 const STOP_MESSAGE = 'Goodbye!';  
14  
15 exports.handler = function(event, context, callback) {  
16   var alexa = Alexa.handler(event, context);  
17   alexa.appId = APP_ID;  
18   alexa.registerHandlers(handlers); //register the handlers function  
19   alexa.execute(); //execute the object  
20 };  
21  
22 const handlers = {  
23   'LaunchRequest': function () { //invoked on launch  
24     this.emit('WelcomeIntent');  
25   },  
26   'WelcomeIntent': function () {  
27     this.emit(':ask', SKILL_WELCOME); //sezáhne se a čeká na odpověď  
28   },  
29   'SayHalloIntent': function () {  
30     var speechOutput = '';  
31     var nameFromBot = '';  
32  
33     nameFromBot = this.event.request.intent.slots.name.value;  
34     speechOutput = "Hey " + nameFromBot + ", nice to meet you!";  
35  
36     this.response.speak(speechOutput);  
37     this.emit(':responseReady');  
38   },  
39   'AMAZON.HelpIntent': function () {  
40     const speechOutput = HELP_MESSAGE; //assign the help message  
41     const reprompt = HELP_REPROMPT; //assign the reprompt message  
42  
43     this.response.speak(speechOutput).listen(reprompt); //setting the help message followed by listen for the reprompt  
44     this.emit(':responseReady'); //send to Alexa  
45   },  
46   'AMAZON.CancelIntent': function () {  
47     this.response.speak(STOP_MESSAGE); //speak Stop message  
48     this.emit(':responseReady');  
49   },  
50   'AMAZON.StopIntent': function () {  
51     this.response.speak(STOP_MESSAGE); //speak Stop message  
52     this.emit(':responseReady');  
53   },  
54 };  
55 };
```

Save

# Skill Interface: test

alexa developer console

Your Skills Greeting Bot Build Test Distribution Certification Analytics

Test is enabled for this skill

Skill I/O Echo Show Display Echo Spot Display Device Log

Alexa Simulator Manual JSON Voice & Tone

English (US) Type or click and hold the mic

ask greeting bot

Hi Martin, who you want to say hello?

say hello to petr

Hey petr, nice to meet you!

Skill I/O

JSON Input

```
16   "applicationId": "amzn1.ask.skill.37c1dce0-522c-4
17   },
18   "user": {
19     "userId": "amzn1.ask.account.AFD5YCFLO3TA77VUQVGY
20   },
21   "device": {
22     "deviceId": "amzn1.ask.device.AG3ROTXNZFESYGSFW4Y
23   },
24   "supportedInterfaces": {}
25 },
26 "apiEndpoint": "https://api.amazonalexa.com",
27 "apiAccessToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Ni
28 }
29 },
30 "request": {
31   "type": "IntentRequest",
32   "requestId": "amzn1.echo-api.request.41404e10-0619-4710-a
33   "timestamp": "2018-09-18T21:35:32Z",
34   "locale": "en-US",
35   "intent": {
36     "name": "SayHelloIntent",
37     "confirmationStatus": "NONE",
38     "slots": {
39       "name": {
40         "name": "name",
41         "value": "petr",
42         "confirmationStatus": "NONE"
43     }
44   }
45 }
```

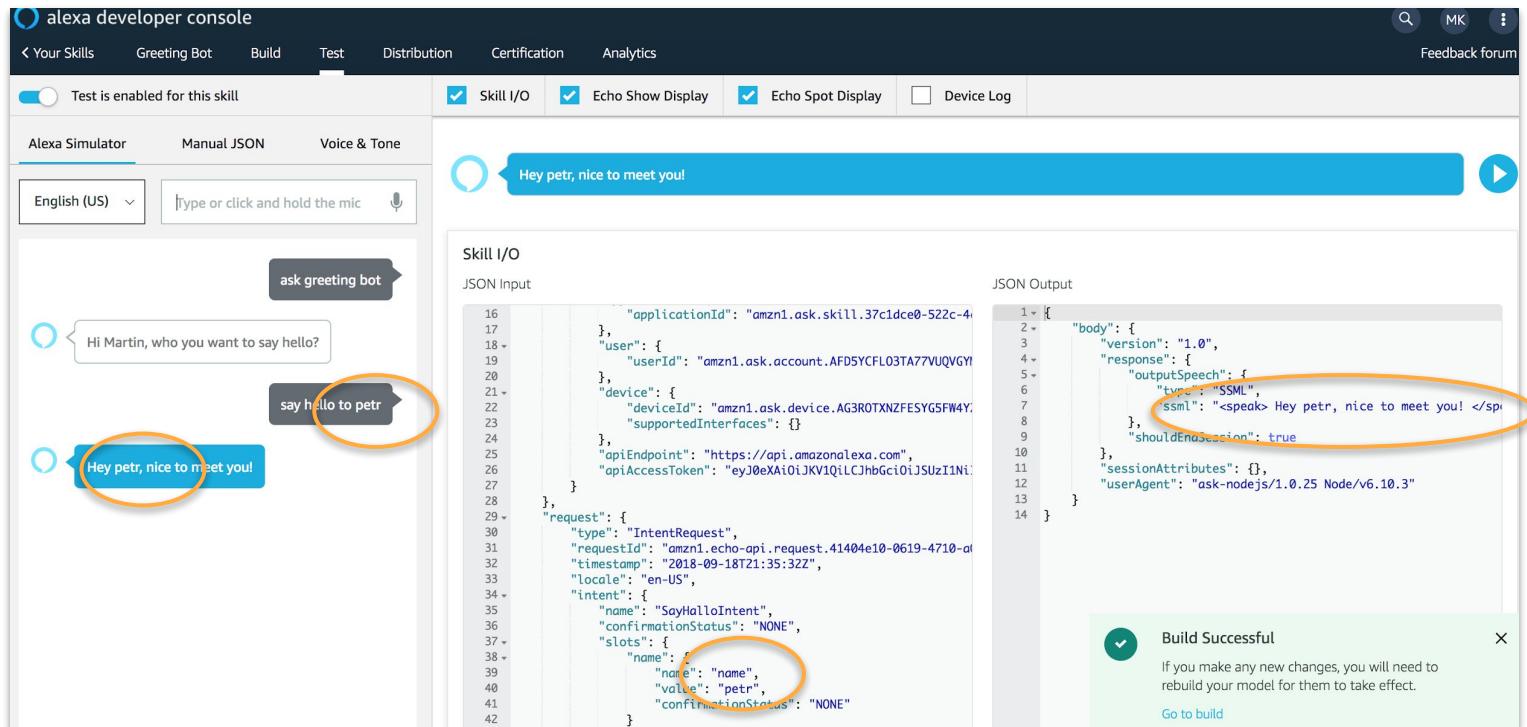
JSON Output

```
1  {
2   "body": {
3     "version": "1.0",
4     "response": {
5       "outputSpeech": {
6         "type": "SSML",
7         "ssml": "<speak> Hey petr, nice to meet you! </sp
8       },
9       "shouldEndSession": true
10    },
11   "sessionAttributes": {},
12   "userAgent": "ask-nodejs/1.0.25 Node/v6.10.3"
13 }
14 }
```

Build Successful

If you make any new changes, you will need to rebuild your model for them to take effect.

Go to build





Alexa, ask Greeting Bot,  
Say hello to Google.

I'm glad you enjoyed meeting me.



# What you cannot do

- Collect sensitive data, GDPR, credit card, healthcare..
- Brands like Nike, McDonald's
- Notifications
- Advertising (either your) Skill
- Actively listen to the mic (make a phone call)

# Testing

## Before launch

- User Testing (by your target groups)

## Production

- Automation test: “CPU is running”
- Is your bot responding correctly?  
Every day?



Money talks: “happy” part

# Publishing + Monetization

- Amazon Pay for Alexa Skills
- Free Skill + In-Skill Purchasing
- Alexa Developer Rewards



# Google & Amazon

No standard

- There is no one “standard” like HTML & web browser
- write, test and run multiple platform (Alexa, Google, Contana, IBM, Siri...)

JOVO

- The Framework for  
Voice App Development
- [www.jovo.tech](http://www.jovo.tech)



# Examples

weather

news

1..10

song

# So let's create a voice bot

I can help you with strategy, conversation design & marketing.



<http://zerozonecreative.com>

Thank you

Martin@Krcek.cz