

Starter Package For Alexa

Tips for Assignment 5

Setup

- There are two main ways to develop skills for Alexa:
 - Alexa Skill Kit
 - Alexa Conversations
- The backend code shouldn't change much for either.
- The following slides mostly give tips on getting started and some possible directions for Alexa Conversation Dialogs

Setup

- Create an account at the developer website:
 - <https://developer.amazon.com/en-US/alexa>
- Alexa Conversations documentation:
 - <https://developer.amazon.com/en-US/docs/alexa/conversations/work-with-alexa-conversations.html>
- Go to “Console” and then on “Create Skill”

Setup

- Host your backend in Alexa-Hosted’s Node.js option

Skill name

133-worldclock

14/50 characters

Brand names are only allowed if you provide proof of rights in the testing instructions or if you use the brand name in a referential manner that doesn't imply ownership (examples of terms that can be added to a brand name for referential usage: unofficial, unauthorized, fan, fandom, for, about).

Default language

This is the language and locale that you will build your skill in. You will be able to add other languages and locales later.

English (US)

▼

More languages can be added to your skill after creation

1. Choose a model to add to your skill

There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are in the Alexa Skills Library and can be added to your skill.

Custom

Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.

SELECTED

Flash Briefing

Give users control of their news feed. This pre-built model lets users control what updates they listen to.

"Alexa, what's in the news?"

Smart Home

Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.

"Alexa, turn on the kitchen lights"

2. Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to the AWS Free Tier and AWS Lambda from the developer console.

Alexa-hosted (Node.js)

Alexa will host skills in your account up to the AWS Free Tier limits and get you started with a Node.js template. You will gain access to AWS Lambda endpoints in all Alexa service regions, a DynamoDB table for data persistence, and S3 for media storage. [Learn more](#)

SELECTED

Alexa-hosted (Python)

Alexa will host skills in your account up to the AWS Free Tier limits and get you started with a Python template. You will gain access to AWS Lambda endpoints in all Alexa service regions, a DynamoDB table for data persistence, and S3 for media storage. [Learn more](#)

Provision your own

Provision your own endpoint and backend resources for your skill. This is recommended for skills that have significant data transfer requirements. You will not gain access to the console's code editor.

Setup

- You can choose the Alexa Conversations Template.
- This will help with some starter code and dialogs

Choose a template to add to your skill

Select a skill template from the list below or import a skill shared by the Alexa community as a public Git repository.

Import skill

Continue with template

Start from Scratch

This skill gets you started with the required intents and with code demonstrating “Hello World” functionality if you are building an Alexa-hosted skill.
[Learn more](#)

By [Alexa](#)

Fact Skill

Build an engaging fact skill about any topic. Alexa will select a fact at random and share it with the user when the skill is invoked. [Learn more](#)

Includes: custom intents

By [Alexa](#)

High-Low Game Skill

Try to guess a target number in a given range and Alexa will tell you if the number she had in mind was higher or lower. [Learn more](#)

Includes: slots, custom intents, data persistence

By [Alexa](#)

Pet Tales Skill

Build a compelling multi-turn conversational audio and visual experience for a user looking for her favorite pet. [Learn more](#)

Includes: APL for Audio, APL, custom intents, data persistence

By [Alexa](#)

Fruit Shop Skill

Build a multi-modal grocery shopping skill using custom and library controls for item lists, shopping cart management, and checkout. [Learn more](#)

Includes: ASK SDK Controls Framework Preview, APL

By [Alexa](#)

Scheduling Skill

Build a skill to allow users to schedule appointments on your calendar, receive email confirmations and reminders. [Learn more](#)

Includes: voice permissions, reminders, API calls, session persistence

By [Dabble Lab](#)

Survey Skill

Build a stand-up or survey skill that uses passcodes to allow only authorized users to provide updates and respond to questions. [Learn more](#)

Includes: using passcodes, API calls, session persistence

By [Dabble Lab](#)

Intro to Alexa Conversations

This skill introduces you to Alexa Conversations by providing basic “Hello World” functionality and generating a voice response from Alexa. [Learn more](#)

Includes: Alexa Conversations Preview, APL, APL for Audio, session persistence

By [Alexa](#)

Weather Bot Skill

Build a conversational weather bot skill that allows users to receive brief weather updates for a given location and date. [Learn more](#)

Includes: Alexa Conversations, APL for Audio, session persistence

By [Alexa](#)

Pizza Ordering Example

An example pizza ordering skill with Alexa Conversations demonstrating user corrections and context carryover. [Learn more](#)

Includes: Alexa Conversations, APL for Audio, session persistence

By [Alexa](#)

Constructing the Alexa Skill

- Skills make use of Slots, that are similar to variable types or definitions
- There are many builtin slot types

Slot Types			
Slot types define how phrases in utterances are recognized and handled as well as the type of data passed between components. In Interaction Model, all intent slots must be assigned a slot type. In Alexa Conversations, all slots, arguments, response types and variables must be assigned a slot type. Learn more about using Slot Types and learn more about using Slot Types with Alexa Conversations.			
<div>+ Add Slot Type</div>		<div>Filter Slot Types</div>	
NAME	SLOT VALUES	SLOT TYPE	ACTIONS
AMAZON.City	0	Built-In	Edit Delete
AMAZON.Color	0	Built-In	Edit Delete
AMAZON.TIME	0	Built-In	Delete
CityTime	-	Custom with properties	Edit Delete

Constructing the Alexa Skill

- A slot can also combine other “primitive” slot types

Slot Types / CityTime

Custom slot types with properties are hierarchical and are only used in Alexa Conversations to pass data between components (e.g. API Definitions, Responses). [Learn more.](#)

Properties (2) ?

Add a new property

NAME	SLOT TYPE	LIST	ACTIONS
time	<div>AMAZON.TIME</div>	<div></div>	Delete
city	<div>AMAZON.City</div>	<div></div>	Delete

Constructing the Alexa Skill

- The template as examples for “Favorite color” dialogs.
- Bellow is an example of a possible Dialog for informing current time

Dialogs / MyTimeNow

Dialogs are sample conversations consisting of turns between the User and Alexa (starting with User). Dialogs require at least one API Invocation. Each turn can be annotated by clicking on the utterance and completing the form. Variables are used through arguments and Response arguments. [Learn more.](#) | [Take a tour](#)

Slot TypesNone

USER:

What is the time?

Invoke APIs

ALEXA:

It is 9:00am now

API Success

Add:

User says

Alexa says

Alexa Response

No issues detected.

Dialog Act - Purpose of Alexa Response ?

API Success

API to Invoke ?

LookupTimeNow

Return Type → Variable ?

AMAZON.TIME

→

amazonTIME0

Response ?

TimeNowResponse

Variable → Arguments ?

amazonTIME0

→

time

Condition (Optional) ?

Add another Dialog Act

Constructing the Alexa Skill

- That dialog is invoking the following API definition
- APIs call backend code handlers

API Definitions / LookupTimeNow

API Definitions define interfaces with your backend service using arguments as inputs and return as output. They are used when annotating Alexa Response turns in a Dialog. [Learn More.](#)

Arguments (0)

Arguments are inputs to API Definitions. They are local in scope and connected to dialogs via variables.

Add Argument

NAME	SLOT TYPE
There are no API arguments available	

Return

Return is the output of the API Definition. It is local in scope and connected to dialogs via variables.

AMAZON.TIME

✕

▼

List: ☐

Setup

- Dialogs make use of utterance sets to teach Alexa what type of utterances to expect

Utterance Sets / AskTimeNow

Utterance Sets are groups of utterances that users may say to Alexa, which can include slots. They are used when

Dialog Act ?

Dialog Act is the purpose of the Utterance Set. [Learn more.](#)

Invoke APIs



Sample Utterances (3)

Sample Utterances are phrases the user may say to Alexa in a response or request (use {} to reference Slots or double click o

[Add Utterance](#)

Time, please?

Whats the time?

What time is it now?

Setup

- Dialogs also need to specify responses

Responses / TimeNowResponse

Responses are how Alexa responds to users in the form of audio and visual elements. They are used when annotating Alexa Response turns in a Dialog. [Learn more](#).

Audio Response (required)

Audio Responses render speech and sound. Edit Audio Response to open APL for Audio editor in a new tab to use advanced features such as sound effects, music, conditions and more. [Learn More](#)

TimeNowResponsePrompt

Clear

Audio Response Name

TimeNowResponsePrompt

!

Save Model and open APL for Audio editor for advanced updates to TimeNowResponsePrompt

Alexa Prompts ⓘ

Add prompt

It is {time.time} right now

Visual Response

Visual Responses render graphics, images, slideshows and video and can be customized for different Echo comparable device types via APL. Edit Visual Response to open APL editor in a new tab. [Learn More](#)

Select a visual response

Refresh List | Clear

Arguments (1)

Arguments are inputs to Responses. They are local in scope and connected to dialogs via variables.

Add Argument

NAME	SLOT TYPES
<div>time</div>	<div>AMAZON.TIME</div>

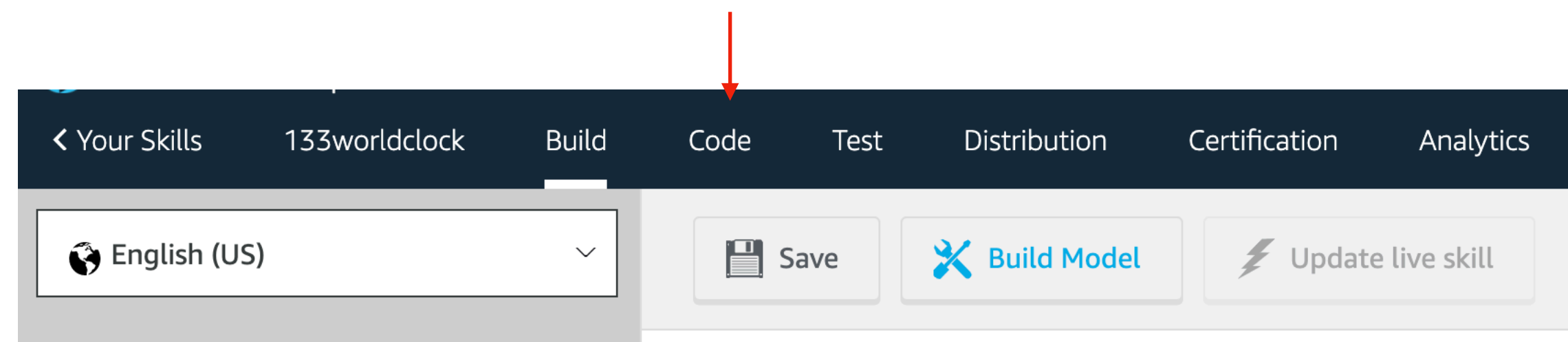
Backend

- Backend runs the code to handle API calls
- The Alexa-hosted backend can be in any of numerous AWS servers
 - The time in the server can be different from the local time of the user talking with the skill
- Alexa supports discovering the device's timezone:

<https://developer.amazon.com/en-US/blogs/alexa/alexa-skills-kit/2019/07/getting-started-with-cake-time-using-the-alexa-settings-api-to-look-up-the-device-time-zone>

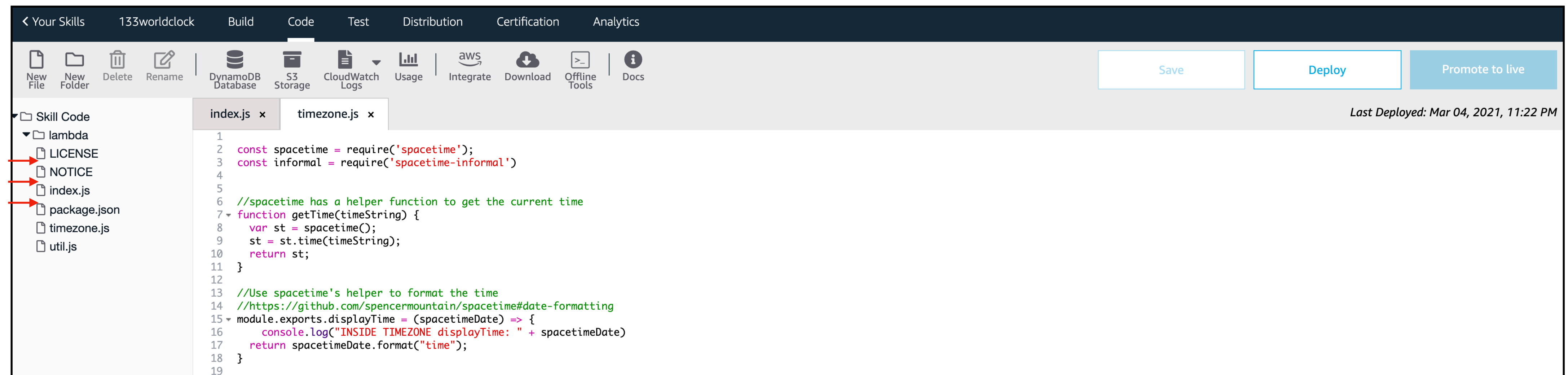
Backend

- We have provided a zip file with a starter code and helper file for your backend functions.
- Navigate to your backend code editor and alter the contents in the provided “lambda” folder accordingly



Backend

- Add spacetime libs to package.json
- Add timezone.js helper file
- Paste index.js content from the one in the starter code



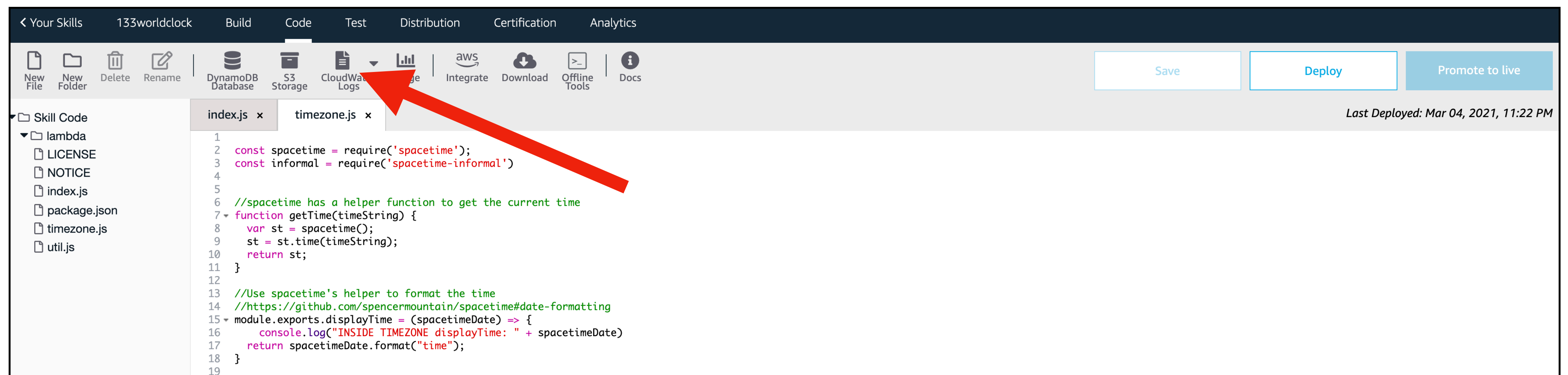
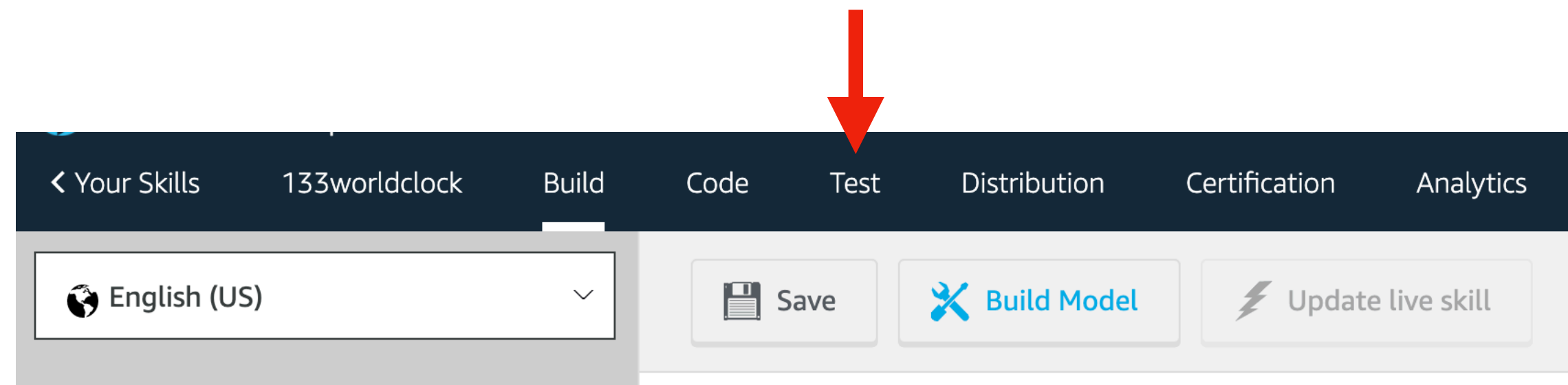
The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with tabs: < Your Skills, 133worldclock, Build, Code, Test, Distribution, Certification, and Analytics. Below this is a toolbar with icons for New File, New Folder, Delete, Rename, DynamoDB Database, S3 Storage, CloudWatch Logs, Usage, Integrate, Download, Offline Tools, and Docs. On the right side of the toolbar are buttons for Save, Deploy, and Promote to live. The left sidebar shows a file tree under 'Skill Code' with a 'lambda' folder containing LICENSE, NOTICE, index.js, package.json, timezone.js, and util.js. The main area displays the code editor for 'index.js'. The code is as follows:

```
1
2  const spacetime = require('spacetime');
3  const informal = require('spacetime-informal')
4
5
6  //spacetime has a helper function to get the current time
7  function getTime(timeString) {
8    var st = spacetime();
9    st = st.time(timeString);
10   return st;
11 }
12
13 //Use spacetime's helper to format the time
14 //https://github.com/spencermountain/spacetime#date-formatting
15 module.exports.displayTime = (spacetimeDate) => {
16   console.log("INSIDE TIMEZONE displayTime: " + spacetimeDate)
17   return spacetimeDate.format("time");
18 }
19
```

At the bottom right of the console, it says "Last Deployed: Mar 04, 2021, 11:22 PM".

Starting the Skill

- Deploy your backend code
- Click on “Test” in the menu
- “console.log()” can be useful in your code. The content is available in the “CloudWatch Logs” page:



Ask for local time

Task 1 example of possible conversation

Skill testing is enabled in: Development

☒ Skill I/O☒ Device Display☐ Device Log

Alexa SimulatorManual JSONVoice & Tone

English (US)Type or click and hold the mic

start my world clock

Welcome to the world clock skill. How can I help?

what time is it now

It is 11:26pm right now

Skill Invocations | Viewing: 1 / 1

JSON Input 1

```
1 {
2   "version": "1.0",
3   "session": {
4     "new": false,
5     "sessionId": "amzn1.echo-api.session.8c1098cc-3445-4647-b892-9c42f5db116f"
6   },
7   "application": {
8     "applicationId": "amzn1.ask.skill.2a15f8f2-954b-4061-99f8-08514210e466"
9   },
10  "user": {
11    "userId": "amzn1.ask.account.AHXCBH5MLNEFCUOMP3NJETP64FNISBWMU5TZR4G0S"
12  },
13  "context": {
14    "Viewports": [
15      {
16        "type": "APL",
17        "id": "main",
18        "shape": "RECTANGLE",
19        "dpi": 160,
20        "presentationType": "STANDARD",
21        "canRotate": false,
22        "configuration": {
23          "current": {
24            "mode": "HUB",
25            "video": {
26              "codecs": [
27                "H_264_42",
28                "H_264_41"
29              ]
30            }
31          }
32        }
33      }
34    ]
35  }
36}
```

JSON Output 1

```
1 {
2   "body": {
3     "version": "1.0",
4     "response": {
5       "directives": [
6         {
7           "type": "Alexa.Presentation.APLA.RenderDocument",
8           "token": "TimeNowResponsePrompt",
9           "document": {
10            "type": "APLA",
11            "version": "0.8",
12            "mainTemplate": {
13              "parameters": [
14                "payload"
15              ],
16              "item": {
17                "type": "Selector",
18                "strategy": "randomItem",
19                "description": "Change 'type' above to try different items",
20                "items": [
21                  {
22                    "type": "Speech",
23                    "contentType": "text",
24                    "content": "It is ${payload.time.time} right now",
25                    "description": "Expand on 'items' array to add more items"
26                  }
27                ]
28              }
29            }
30          }
31        }
32      ]
33    }
34  },
35  "datasources": {
36    "payload": {
37      "time": "11:26pm"
38    }
39  }
40}
```

Convert local time to another Location

Task 2 example of possible conversation

Skill testing is enabled in: Development

☒ Skill I/O ☒ Device Display ☐ Device Log

Alexa Simulator Manual JSON Voice & Tone

English (US) Type or click and hold the mic

start my world clock

Welcome to the world clock skill. How can I help?

lookup time

What is the city's name?

tokyo

It is 4:23pm in tokyo

Skill Invocations | Viewing: 1 / 1

JSON Input 1

```
1 {
2   "version": "1.0",
3   "session": {
4     "new": false,
5     "sessionId": "amzn1.echo-api.session.24f5b5d2-1e59-465d-9a25-a7aa6e55f3e7",
6     "application": {
7       "applicationId": "amzn1.ask.skill.2a15f8f2-954b-4061-99f8-08514210e461"
8     },
9     "user": {
10      "userId": "amzn1.ask.account.AHXCBSMLNEFCUOMP3NJETP64FNISBWMU5TZR4GO"
11    }
12  },
13  "context": {
14    "Viewports": [
15      {
16        "type": "APL",
17        "id": "main",
18        "shape": "RECTANGLE",
19        "dpi": 160,
20        "presentationType": "STANDARD",
21        "canRotate": false,
22        "configuration": {
23          "current": {
24            "mode": "HUB",
25            "video": {
26              "codecs": [
27                "H_264_42",
28                "H_264_41"
29              ]
30            }
31          }
32        }
33      }
34    ]
35  }
36}
```

JSON Output 1

```
1 {
2   "body": {
3     "version": "1.0",
4     "response": {
5       "directives": [
6         {
7           "token": "CityTimeSuccessPrompt",
8           "document": {
9             "type": "APLA",
10            "version": "0.8",
11            "mainTemplate": {
12              "parameters": [
13                "payload"
14              ],
15              "item": {
16                "type": "Selector",
17                "strategy": "randomItem",
18                "description": "Change 'type' above to try different items",
19                "items": [
20                  {
21                    "type": "Speech",
22                    "contentType": "text",
23                    "content": "It is ${payload.cityTime.time} in ${payload.cityTime.location}",
24                    "description": "Expand on 'items' array to add more items"
25                  }
26                ]
27              }
28            }
29          }
30        },
31        "datasources": {
32          "cityTime": {
33            "type": "Text",
34            "content": "It is 4:23pm in tokyo"
35          }
36        }
37      }
38    }
39  }
40}
```