

Alexa Skill

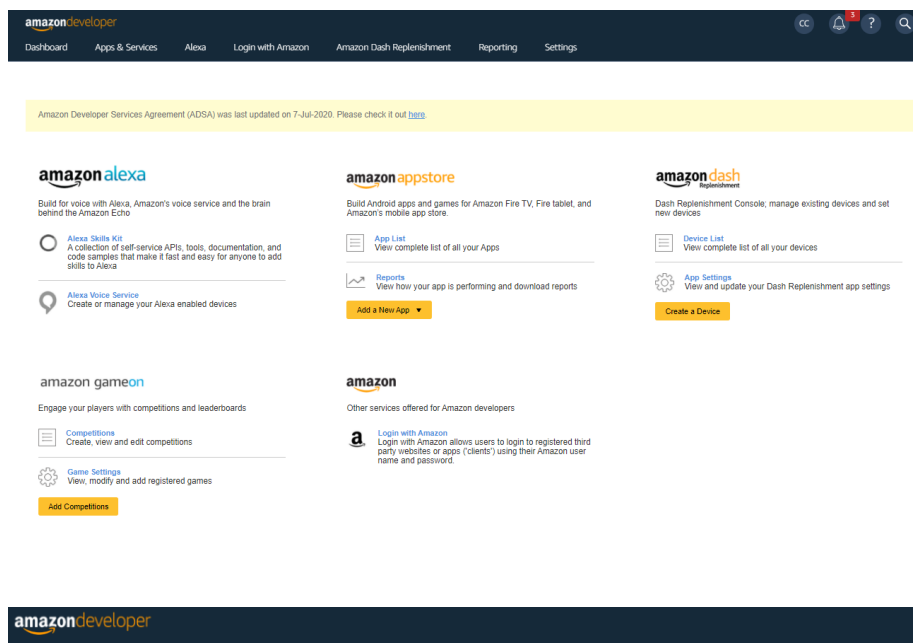
Primeros Pasos

Puedes encontrar Cursos completos y gratuitos sobre Alexa y otras tecnologías en español en:

<https://plataforma.keepcoding.io/courses?query=alexa>

Preparando el Entorno de trabajo


1. Entramos en Amazon Developers y nos creamos una cuenta si no tenemos



The screenshot shows the Amazon Developer console dashboard. At the top is a navigation bar with links: Dashboard, Apps & Services, Alexa, Login with Amazon, Amazon Dash Replenishment, Reporting, and Settings. Below the navigation bar is a yellow banner with the text: "Amazon Developer Services Agreement (ADSA) was last updated on 7-Jul-2020. Please check it out [here](#)". The main content area is divided into several sections:

- amazon alexa**: Build for voice with Alexa, Amazon's voice service and the brain behind the Amazon Echo. It includes links for **Alexa Skills Kit** (A collection of self-service APIs, tools, documentation, and code samples that make it fast and easy for anyone to add skills to Alexa) and **Alexa Voice Service** (Create or manage your Alexa enabled devices).
- amazon appstore**: Build Android apps and games for Amazon Fire TV, Fire tablet, and Amazon's mobile app store. It includes links for **App List** (View complete list of all your Apps) and **Reports** (View how your app is performing and download reports). There is a yellow button labeled "Add a New App".
- amazon dash**: Dash Replenishment Console; manage existing devices and set new devices. It includes links for **Device List** (View complete list of all your devices) and **App Settings** (View and update your Dash Replenishment app settings). There is a yellow button labeled "Create a Device".
- amazon gameon**: Engage your players with competitions and leaderboards. It includes links for **Competitions** (Create, view and edit competitions) and **Game Settings** (View, modify and add registered games). There is a yellow button labeled "Add Competitions".
- amazon**: Other services offered for Amazon developers. It includes a link for **Login with Amazon** (Login with Amazon allows users to login to registered third party websites or apps ("clients") using their Amazon user name and password).

At the bottom of the dashboard is a dark blue bar with the "amazon developer" logo.



Sign up to . . .

- Publish your apps and skills across millions of devices in over 236 countries and territories.
- Build for natural voice experiences.
- Start generating revenue in a whole new way with Amazon's unique services and APIs.

Amazon Developer Registration

Developer details

First name *

 ✓

Last name *

 ✓

Country / Region *

 ▼

Company name *

Company phone number *

▼

Conceptos fundamentales

ASK, AVS

- ASK es una colección de auto-servicios APIs, herramientas, documentación y ejemplos de código que hace rápido y fácil añadir skills a Alexa.
- Amazon permite a los desarrolladores construir y publicar skills para Alexa usando el Kit de Skills de Alexa.
- Alexa Voice Service (AVS) permite a terceros añadir control de voz inteligente a cualquier producto conectado que tenga un micrófono y un altavoz.

Intents, Utterances y Slots

Intents: las acciones que se van a permitir hacer al usuario,

Utterances: las diferentes expresiones que podrá usar el usuario para poder invocar las acciones,

Slots: las variables que el usuario incluirá en las expresiones

Arquitectura

La **arquitectura** propuesta por **Alexa** para que implementemos una Skill se basa en utilizar un modelo de interacción donde podemos definir todos los Intents (intenciones) junto con los Slots (entidades) extraídas de las frases del usuario.

Alexa Skills Kit: Processing a request

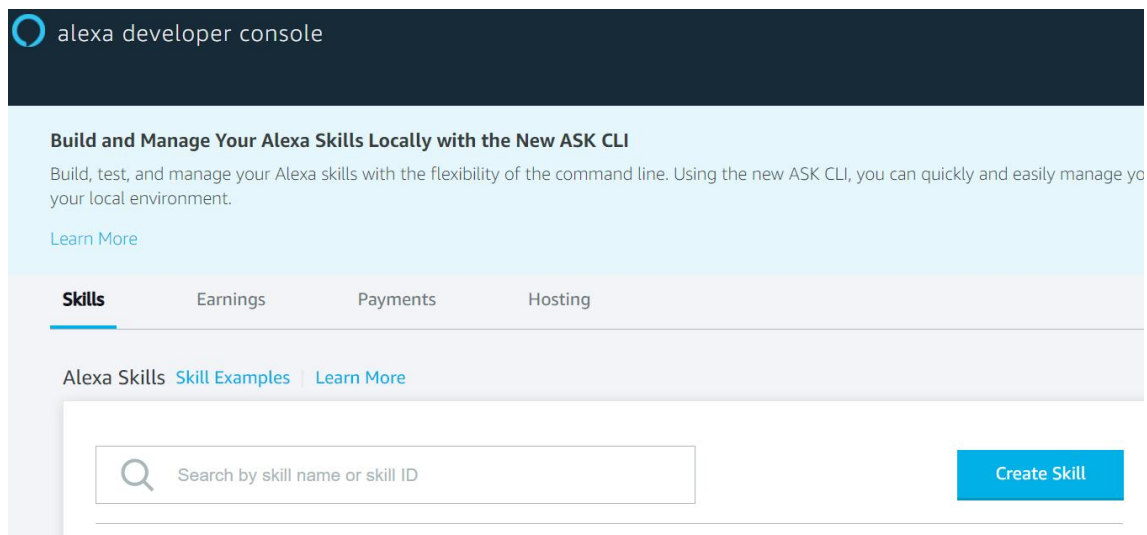


Primer Ejemplo

Para este primer ejemplo Indicamos un **nombre**, ejemplos: historias del fútbol, frases célebres, curiosidades de Tenerife, etc...

Seleccionamos el idioma, en principio español de España **Spanish (ES)**, aunque podría ser español de américa u otro idioma.

Seleccionamos un modelo de skill que pueden ser personalizado, tarjetas flash, Smart home o Video, en principio escogemos personalizado (**Custom**)



The screenshot shows the Alexa Developer Console. At the top, there's a dark blue header with the Alexa logo and 'alexa developer console'. Below it, a light blue banner reads 'Build and Manage Your Alexa Skills Locally with the New ASK CLI' with a 'Learn More' link. A navigation bar contains 'Skills', 'Earnings', 'Payments', and 'Hosting', with 'Skills' being the active tab. Below the navigation bar, there's a section for 'Alexa Skills' with links to 'Skill Examples' and 'Learn More'. A search bar is present with the placeholder 'Search by skill name or skill ID' and a 'Create Skill' button on the right.

Skill name

Skill name must have at least 2 characters.

0/50 characters

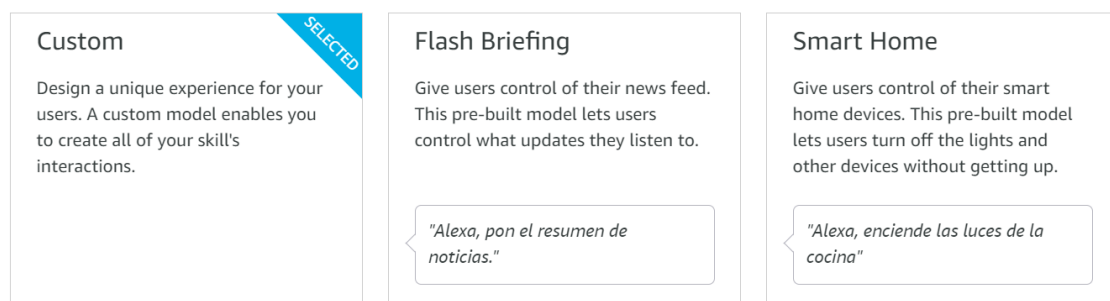
Default language

Spanish (ES) ▼

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 interaction models that contain a package of intents and utterances that you can add to your skill.

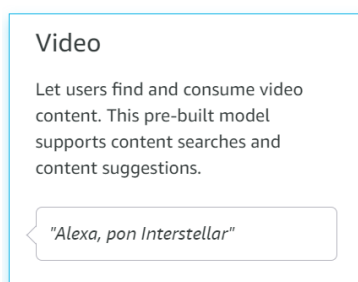


The image shows three cards for selecting a model. The 'Custom' card is highlighted with a blue 'SELECTED' banner. Each card has a title, a description, and a sample utterance in a speech bubble.

Model	Description	Sample Utterance
Custom (SELECTED)	Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.	
Flash Briefing	Give users control of their news feed. This pre-built model lets users control what updates they listen to.	"Alexa, pon el resumen de noticias."
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, enciende las luces de la cocina"

Finalmente seleccionamos el método para alojar los recursos del Backend, para ello se puede crear un servidor propio alojado donde el desarrollador quiera y mantenga, pero está disponible Alexa-Hosted que proporciona un AWS Lambda gestionado por Amazon y preconfigurado para funcionar en las skills de alexa, escogemos **Alexa-Hosted (Node.js)**

AWS Lambda es un servicio de pago de Amazon, pero para Skills de Alexa proporciona 5GB de almacenamiento gratuito y 15GB de transferencia de datos, esto le permite ejecutar código sin aprovisionar ni administrar servidores. Lambda puede ejecutar código para casi cualquier tipo de aplicación o servicio backend sin tener que realizar tareas de administración. Solo se tiene que cargar el código y Lambda se encargará de todo lo necesario para ejecutar y escalar el código con alta disponibilidad.



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 our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

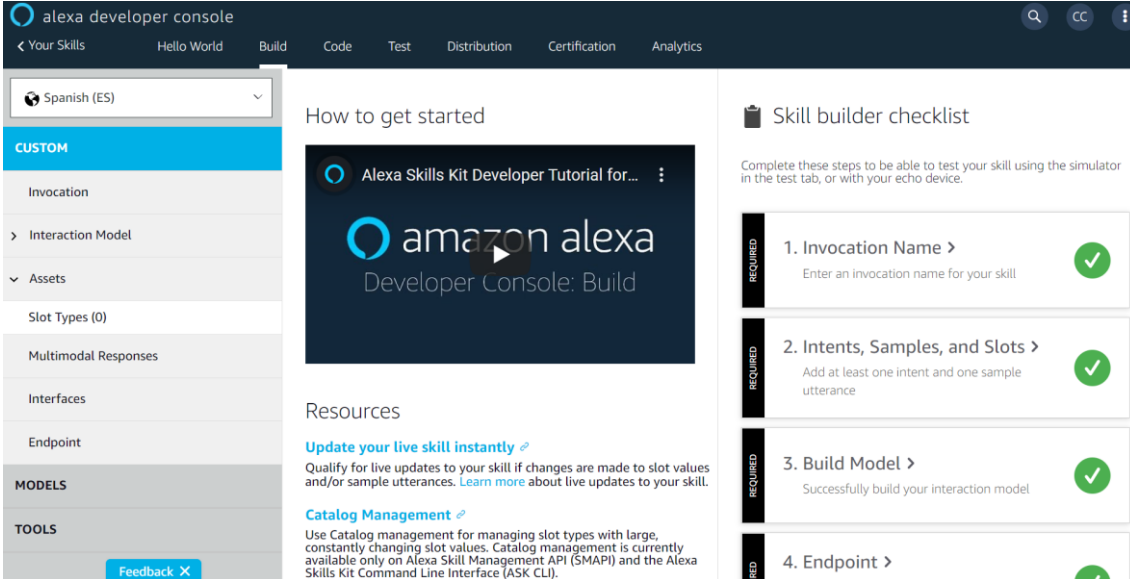
Alexa-Hosted (Node.js) SELECTED	Alexa-Hosted (Python)	Provision your own
Alexa will host skills in your account up to the AWS Free Tier limits and get you started with a Node.js template. You will gain access to an AWS Lambda endpoint, 5 GB of media storage with 15 GB of monthly data transfer, and a table for session persistence. Learn more	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 an AWS Lambda endpoint, 5 GB of media storage with 15 GB of monthly data transfer, and a table for session persistence. Learn more	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.
Feedback X		

Nota: El nombre de la skill y el invocation name (palabra o conjunto de palabras usado para activar la skill) no tiene por qué ser igual.

Para esta práctica desarrollaremos una skill de curiosidades (curiosidades espaciales) pero cada uno puede seleccionar el tema que más le atraiga (deportes, emprendedores, música, cine, etc...)

Una vez seleccionado y rellenado todos los campos anteriores se creará el front-end y back-end por defecto con una skill "hola mundo".

Si a esta skill le preguntas con un hola mundo, te responderá.

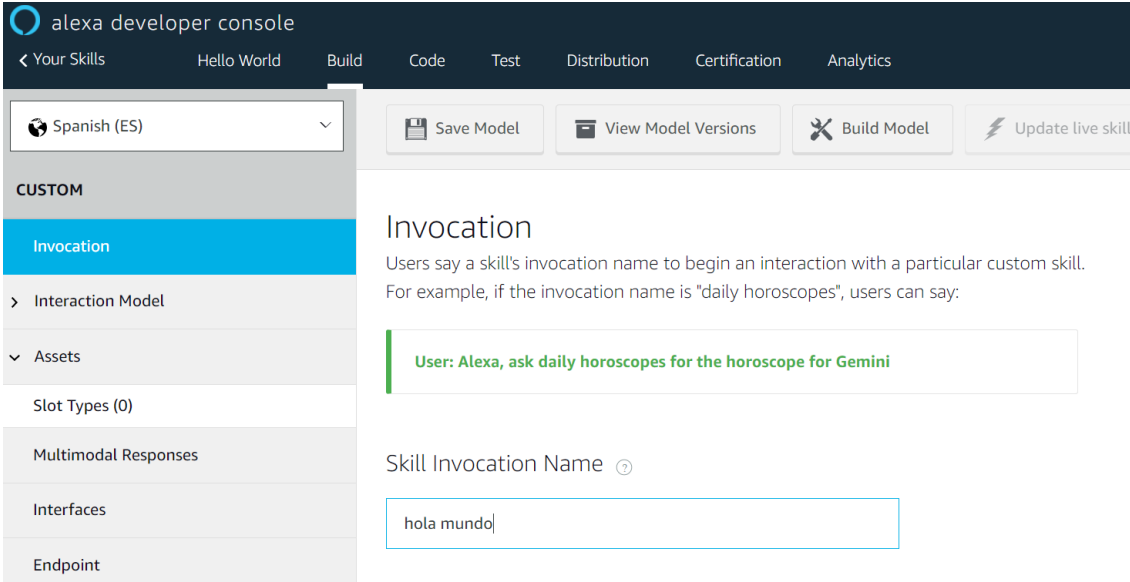


Se puede **probar** desde la pestaña **test**.

“Alexa abre curiosidades espaciales”

Build (Front-end)

El primer paso será desarrollar el front-end o lo que es lo mismo, las frases que alexa va a reconocer dentro de tu skill. La primera será la frase o palabra de invocación.

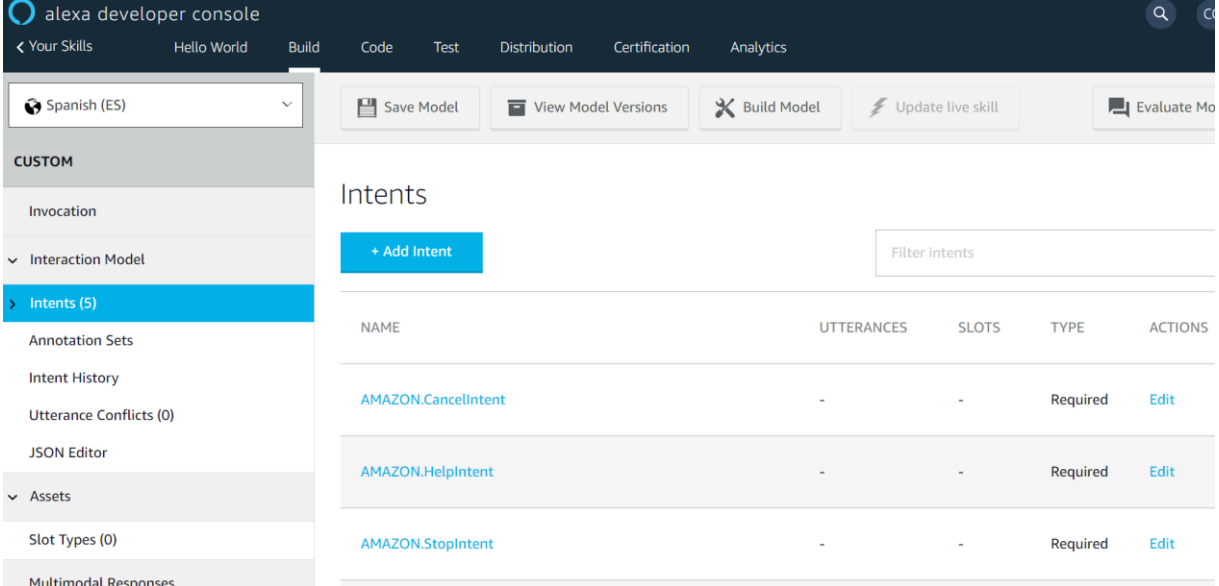


Nota: Siempre que se modifica algo se debe grabar para no perder los datos si cerramos sin querer la pestaña y construir (solo si se va a probar).

A continuación, dentro de la pestaña “*Interaction Model*” seleccionamos “*Intents*”

En esta pestaña hay una serie de intents obligatorios o required que ya vienen por defecto:

Amazon.CancelIntent, HelpIntent, StopIntent, NavegateHomeIntent y otros que se pueden crear NextIntent, NoIntent, YesIntent, GetNewFactIntent, etc...



The screenshot shows the Alexa Developer Console interface. The left sidebar has a menu with 'Intents (5)' selected. The main area displays a table of default intents. The table has columns for NAME, UTTERANCES, SLOTS, TYPE, and ACTIONS. Three intents are listed: AMAZON.CancelIntent, AMAZON.HelpIntent, and AMAZON.StopIntent, all of type 'Required'.

NAME	UTTERANCES	SLOTS	TYPE	ACTIONS
AMAZON.CancelIntent	-	-	Required	Edit
AMAZON.HelpIntent	-	-	Required	Edit
AMAZON.StopIntent	-	-	Required	Edit

GetNewFactIntent

Para este ejemplo cambiaremos el intent que la plataforma nos crea por defecto para la skill Hola Mundo HelloWorldIntent por GetNewFactIntent.

Nota: si quieren antes de realizar estos cambios vayan a la pestaña Test y prueben la skill por defecto HolaMundo

El objetivo de este intent que hemos actualizado y que puede tener otro nombre es el de interactuar con la skill una vez abierta, de esta manera se indicarán las palabras o frases que, una vez se haya seleccionado la skill con el invocation name, esté preparado para responder a dichas interacciones.

Intents / GetNewFactIntent

Sample Utterances (9) ⓘ

What might a user say to invoke this intent?

qué hay

hola que tal

hola

como estás

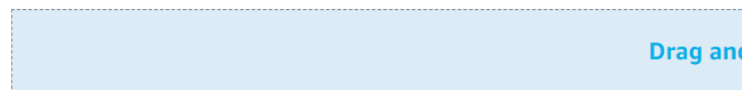
dime una curiosidad



En la pestaña JSON Editor se puede ver y modificar el código JSON que se genera a partir de las pestañas anteriores.

JSON Editor

[Click here](#) to learn more about the schema definition for inter

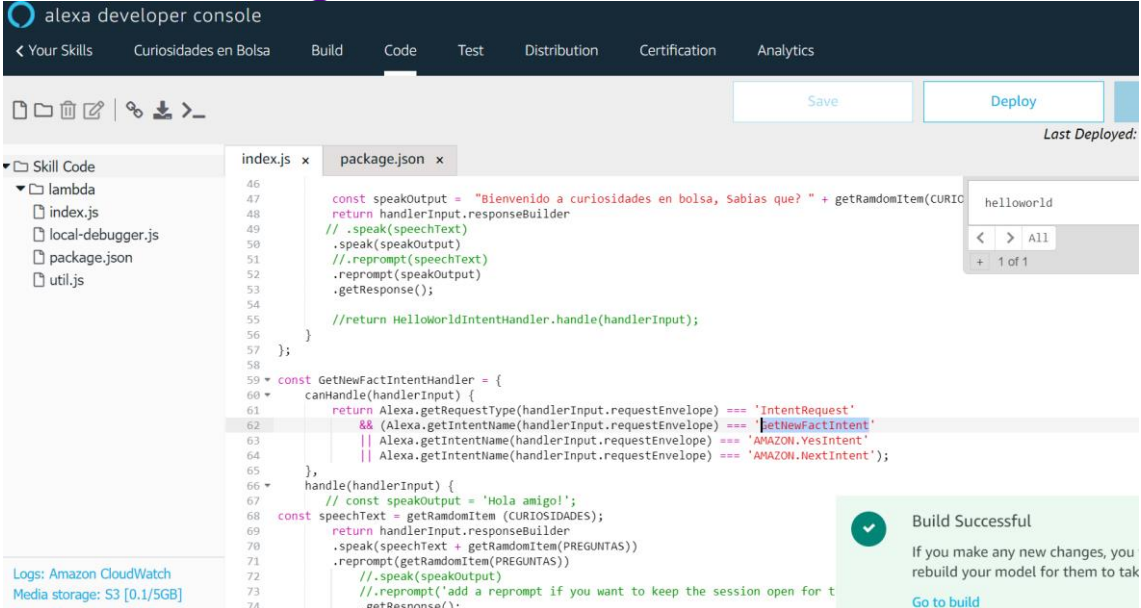


```
1 {  
2   "interactionModel": {  
3     "languageModel": {  
4       "invocationName": "curiosidades en bolsa",  
5       "intents": [  
6         {  
7           "name": "AMAZON.CancelIntent",  
8           "samples": []  
9         },  
10        {  
11          "name": "AMAZON.HelpIntent",  
12          "samples": []  
13        },  
14        {  
15          "name": "AMAZON.StopIntent",  
16          "samples": [  
17            "basta",  
18            "terminar"  
19          ]  
20        },  
21        {  
22          "name": "GetNewFactIntent",  
23          "slots": [],  
24          "samples": [  
25            "qué hay",  
26            "hola que tal",  
27            "hola",  
28            "como estás",  
29            "dime una curiosidad",  
30            "una curiosidad",  
31            "quiero una curiosidad",
```

Código (Back-end)

A continuación pasamos al Back-end, se debe tener en cuenta que el desarrollo de una skill es un proceso iterativo donde se puede ir al back-end y volver al front-end todas las veces que sea necesario, e ir probando en la pestaña **Test** las diferentes implementaciones que se vayan desarrollando.

Para el desarrollo de la skill propuesta se debe cambiar en el código todo lo que tenga HelloWorldIntent por GetNewFactIntent



alex developer console

< Your Skills Curiosidades en Bolsa Build Code Test Distribution Certification Analytics

Save Deploy

Last Deployed:

helloworld

< > All

+ 1 of 1

Build Successful

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

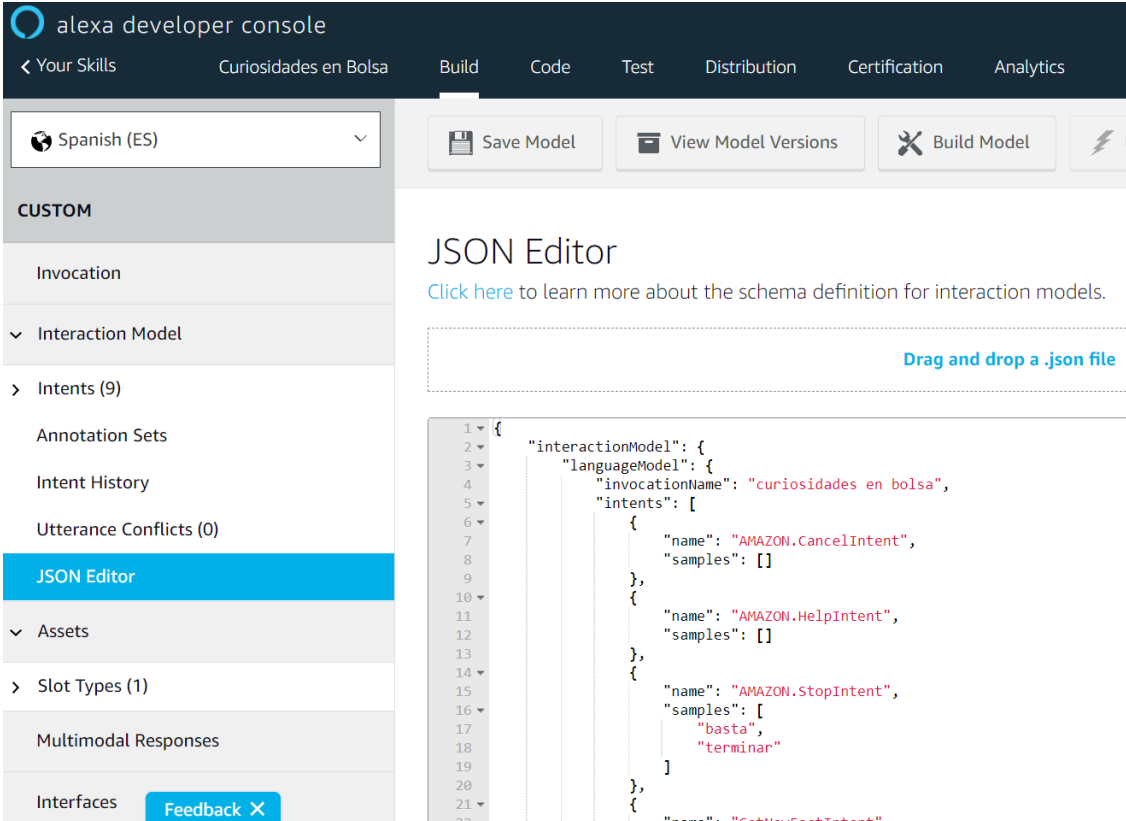
Go to build

El desarrollo de una skill es un proceso iterativo donde es aconsejable “Guardar, construir y probar” con frecuencia.

Todo el código del ejemplo para “Curiosidades espaciales” lo pueden encontrar aquí:

<https://gist.github.com/germanviscuso/6f80f5b6ceb7b9674788dc3971adc932>

Se pueden modificar o ampliar los intents por defecto (Help, Cancel, Yes, ...)



alex developer console

< Your Skills Curiosidades en Bolsa Build Code Test Distribution Certification Analytics

Spanish (ES)

Save Model View Model Versions Build Model

CUSTOM

Invocation

Interaction Model

Intents (9)

Annotation Sets

Intent History

Utterance Conflicts (0)

JSON Editor

Assets

Slot Types (1)

Multimodal Responses

Interfaces Feedback

JSON Editor

Click here to learn more about the schema definition for interaction models.

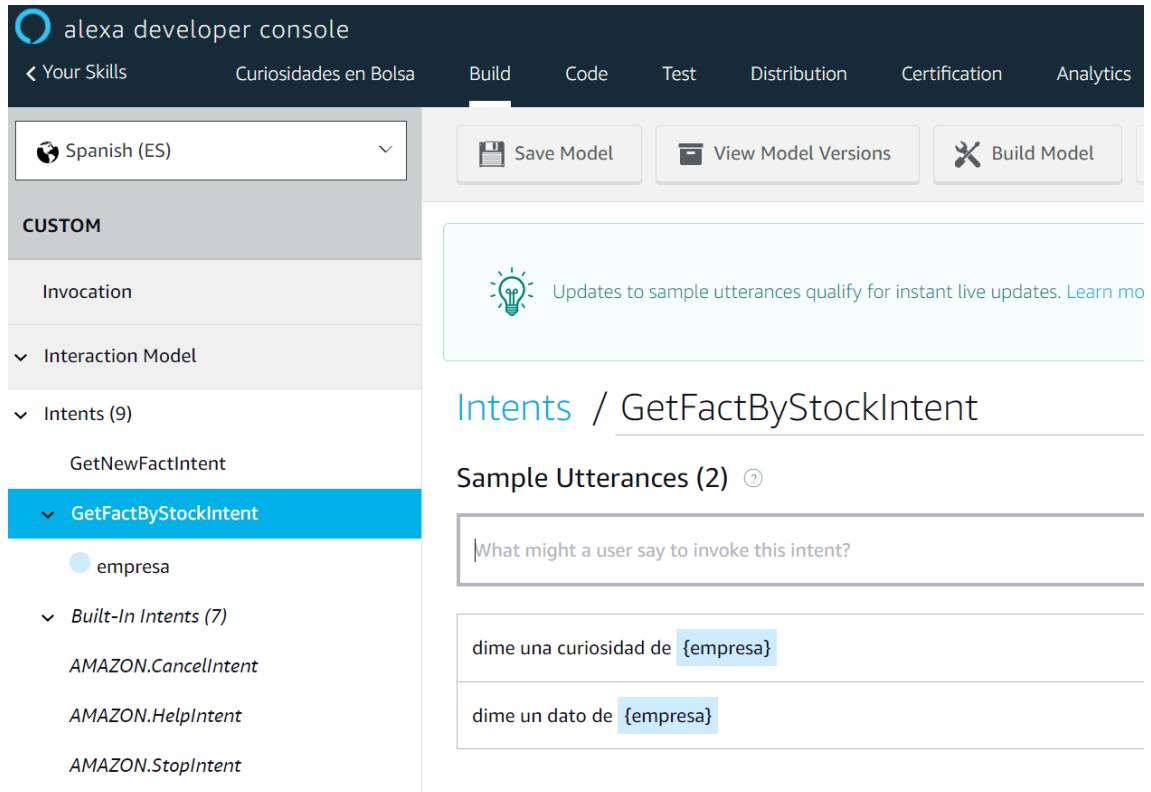
Drag and drop a .json file

```

1 {
2   "interactionModel": {
3     "languageModel": {
4       "invocationName": "curiosidades en bolsa",
5       "intents": [
6         {
7           "name": "AMAZON.CancelIntent",
8           "samples": []
9         },
10        {
11          "name": "AMAZON.HelpIntent",
12          "samples": []
13        },
14        {
15          "name": "AMAZON.StopIntent",
16          "samples": [
17            "basta",
18            "terminar"
19          ]
20        },
21        {
22          "name": "GetNewFactIntent".

```


Creamos un nuevo Intent



alexa developer console

< Your Skills Curiosidades en Bolsa Build Code Test Distribution Certification Analytics

Spanish (ES)

CUSTOM

Invocation

Interaction Model

Intents (9)

GetNewFactIntent

GetFactByStockIntent

empresa

Built-In Intents (7)

AMAZON.CancelIntent

AMAZON.HelpIntent

AMAZON.StopIntent

Save Model View Model Versions Build Model

Updates to sample utterances qualify for instant live updates. [Learn more](#)

Intents / GetFactByStockIntent

Sample Utterances (2) ?

What might a user say to invoke this intent?

dime una curiosidad de {empresa}

dime un dato de {empresa}

Definir un slot en el intent

Intent Slots (1) ?

ORDER ?	NAME ?	SLOT TYPE ?	ACTIONS
1	empresa	AMAZON.Corporation	Edit Dialog Delete
2	Create a new slot	+ Select a slot type	Edit Dialog Delete

Se puede elegir distinto tipo de datos en el Slot Type que en tipo de datos, si lo que quieres añadir no está, siempre puedes crearlo, por ejemplo TipoPlanetas con valores (Mercurio, Venus, Tierra, Marte...)

Nota: en este caso de definir un slot no predeterminado habrá que traducir dicho slot a los diferentes idiomas a los que se le de soporte a la skill.

A continuación en code -> index.js

LaunchRequestHandler: este handler (manejador) es el que obtiene el control cuando se llama al invocation name, si se llama a:

```
return GetNewFactIntentHandler.handle(handlerInput);
```

se le da el control al GetNewFactIntentHandler que se ha creado, y será el encargado de armar la respuesta y también puede permitir futuras interacciones añadiendo

```
|| Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.YesIntent'  
  
|| Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.NextIntent');
```

Con la función getRandomItem obtendremos un elemento al azar de curiosidades.

```
function getRandomItem(array) {  
    return array[Math.floor(Math.random()*array.length)]  
}  
  
const CURIOSIDADES = [  
    "Warren Buffet es el unico multimillonario entre los 10 primeros puestos que no se dedica a la tecnología. ",  
    "Elon Musk ya es el tercero más rico. ",  
    "Jeff Bezos ha sido el hombre más millonario del mundo los tres últimos años. ",  
    "Microsoft es la única compañía que estaba entre las 5 mayores compañías en 2000 y sigue todavía entre las 5 mayores. ",  
    "Tesla vale mas en bolsa que Volkswagen, Audi y Mercedes, juntas. ",  
    "Amazon vale más en bolsa que el Dax completo. ",  
    "Amancio Ortega fué el hombre más millonario del mundo, ya no está ni entre los 10 primeros. ",  
    "El Standar and poors ya superó los máximos de febrero del 2000. ",  
    "El origen de la Bolsa de Comercio de Nueva York se debe a un árbol que había en Wall Street en el que se reunían financieros y especuladores. ",  
    "Fue en el año 1460 cuando se inició la Bolsa de Amberes (Bélgica). Amsterdam fue la segunda ciudad en constituirse. ",  
    "Tres Bolsas en Colombia. El país latinoamericano llegó a poseer tres Bolsas diferentes: Bolsa de Bogotá (1928), Bolsa de Medellín (1961) y Bols:  
    "La Bolsa de Valores apuesta por las mujeres porque está demostrado estadísticamente que son mejores analizando mercados. ",  
    "La palabra 'bolsa' tiene su origen en un edificio que perteneció a una familia noble, de apellido Van Der Buerse, en la ciudad europea de Bruja:  
    "La acción más cara disponible en el NYSE es Berkshire Hathaway, El precio actual por acción es de 304.000 dólares. "  
];
```

Y se añade un valor al azar de preguntas

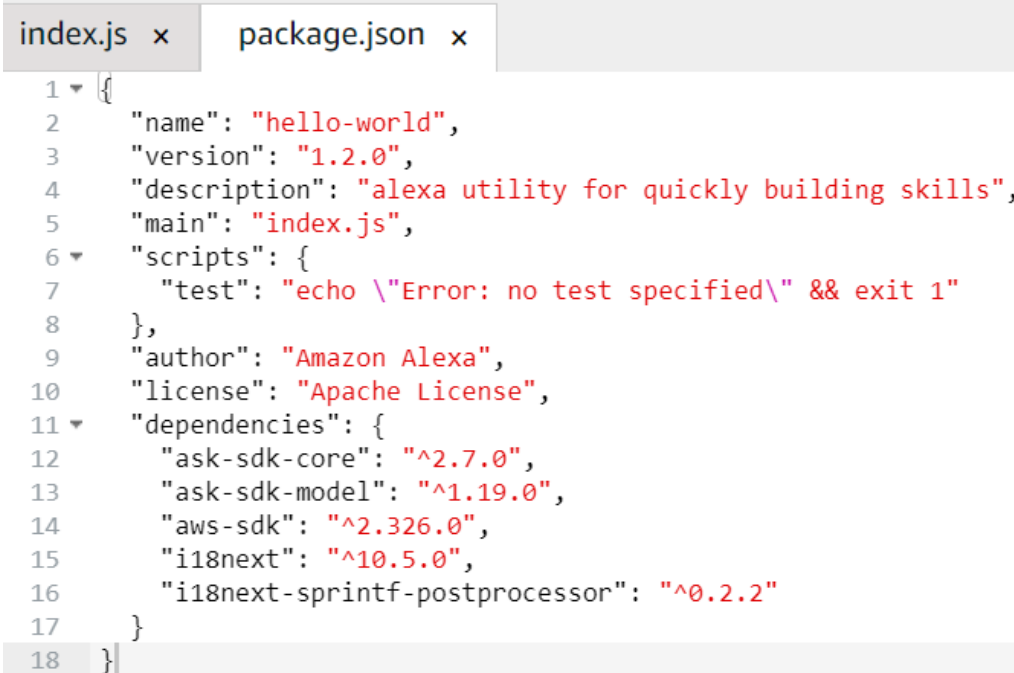
```
const PREGUNTAS = [  
    "quieres otra?",  
    "quieres otra curiosidad?",  
    "te gustaría saber más?",  
    "te gustaría saber otra?",  
    "otra?",  
    "quieres saber la siguiente?",  
    "quieres saber otra?",  
    "quieres saber otra curiosidad?",  
    "te digo otra?",  
    "te digo otra curiosidad?",  
    "te digo la siguiente curiosidad?",  
    "quieres una nueva?",  
    "una más?"  
];
```

El reprompt permite dejar la sesión abierta tras realizar una respuesta. Hay 8 segundos para responder en dispositivo real, en pruebas queda a la espera de forma indefinida, incluso se puede cambiar el backend y se puede continuar con la interacción en testeo.

```
.speak(speechText + getRandomItem(PREGUNTAS))  
.reprompt(getRandomItem(PREGUNTAS))
```

El `SessionEndedRequestHandler` se activa cuando se cierra la sesión por lo que es ideal para cerrar una base de datos que se haya abierto u otros servicios.

En `package.json` se pueden añadir dependencias, y al hacer un deploy internamente se ejecuta un `npm install` dependencia, y ya las dependencias estarán accesibles.



```
index.js x package.json x
1 {
2   "name": "hello-world",
3   "version": "1.2.0",
4   "description": "alexa utility for quickly building skills",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "author": "Amazon Alexa",
10  "license": "Apache License",
11  "dependencies": {
12    "ask-sdk-core": "^2.7.0",
13    "ask-sdk-model": "^1.19.0",
14    "aws-sdk": "^2.326.0",
15    "i18next": "^10.5.0",
16    "i18next-sprintf-postprocessor": "^0.2.2"
17  }
18 }
```

`ErrorHandler` salta cada vez que haya un error, se puede informar al usuario que hay un error, por ejemplo, "Perdón, hay un error, inténtalo otra vez"

Y con `console.log('Error: ${error.message}');` se puede loggear el error, útil para depurar.

Exportar los handlers para que el servicio de Alexa lo pueda ver.

Nota: es importante el orden, si uno se activa, no se activarán los siguientes.

```

/**
 * This handler acts as the entry point for your skill, routing all request and response
 * payloads to the handlers above. Make sure any new handlers or interceptors you've
 * defined are included below. The order matters - they're processed top to bottom
 */
exports.handler = Alexa.SkillBuilders.custom()
    .addRequestHandlers(
        LaunchRequestHandler,
        GetNewFactIntentHandler,
        HelpIntentHandler,
        CancelAndStopIntentHandler,
        FallbackIntentHandler,
        SessionEndedRequestHandler,
        IntentReflectorHandler)
    .addErrorHandlers(
        ErrorHandler)
    .addRequestInterceptors(
        LocalizationInterceptor,
        LoggingRequestInterceptor)
    .addResponseInterceptors(
        LoggingResponseInterceptor)
    // .addRequestInterceptors(LoggingRequestInterceptor, LocalizationInterceptor)
    // .addResponseInterceptors(LoggingResponseInterceptor)
    // .withCustomUserAgent('sample/hello-world/v1.2')
    .lambda();

```

En la pestaña distribution se puede poner un nombre y una descripción distinta para la aplicación en cada Idioma, eso puede ayudar al posicionamiento de la skill.

Reminders. Api permite enviar notificaciones a Alexa

Proactive events notifications, enciende una luz el dispositivo y el usuario le puede pedir que lea las notificaciones.

Diseñar una skill de Alexa propia





Universidad
de La Laguna

Inspecciona la interacción

