Build Skills with the Alexa Skills Kit

Alexa provides a set of built-in capabilities, referred to as *skills*. For example, Alexa's abilities include playing music from multiple providers, answering questions, providing weather forecasts, and querying Wikipedia.

The Alexa Skills Kit lets you teach Alexa *new skills*. Customers can access these new abilities by asking Alexa questions or making requests. You can build skills that provide users with many different types of abilities. For example, a skill might do any one of the following:

* Look up answers to specific questions (*"Alexa, ask tide pooler for the high tide today in Seattle."*)
* Challenge the user with puzzles or games (*"Alexa, play Harry Potter quiz."*)
* Control lights and other devices in the home (*"Alexa, turn on the living room lights."*)
* Provide audio or text content for a customer's flash briefing (*"Alexa, give me my flash briefing"*)

What Kind of Skill Do You Want to Create?

The first step in building a new skill is to decide *what your skill will do*. The functionality you want to implement determines how your skill integrates with the Alexa service and what you need to build. The Alexa Skills Kit supports building different types of skills.

| To create ... | Use this skill type |
| --- | --- |
| A skill that can handle just about any type of request.  For example:   * Look up information from a web service * Integrate with a web service to order something (order a car from Uber, order a pizza from Domino's Pizza) * Interactive games * Just about anything else you can think of | [Custom skill](https://developer.amazon.com/docs/custom-skills/understanding-custom-skills.html) (*custom interaction model*)  You define the requests the skill can handle (*intents*) and the words users say to invoke those requests (*utterances*). |
| A skill that lets a user control and query cloud-enabled smart home devices such as lights, door locks, cameras, thermostats and smart TVs.  For example:   * Turn lights on and off * Change the brightness of dimmable lights * Change the color or color temperature of a tunable light * Change the temperature on a thermostat * Query a lock to see if it is currently locked * Ask for a smart home camera feed * Change the volume on a speaker * Change the channel of a TV | [Smart Home Skill API](https://developer.amazon.com/docs/smarthome/understand-the-smart-home-skill-api.html) (*pre-built model*)  The Smart Home Skill API defines the requests the skill can handle (*device directives*) and the words users say to invoke those requests (*utterances*). |
| A skill that lets a user control cloud-enabled video services.  For example:   * Play a movie * Find a TV show * Change a channel * Pause, rewind, or fast forward video content | [Video Skill API](https://developer.amazon.com/docs/video/understand-the-video-skill-api.html) (*pre-built model*)  The Video Skill API defines the requests the skill can handle (device directives) and the words users say to invoke those requests (utterances). |
| A skill that provides original content for a customer's flash briefing. | [Flash Briefing Skill API](https://developer.amazon.com/docs/flashbriefing/understand-the-flash-briefing-skill-api.html) (*pre-built model*)  The Flash Briefing Skill API defines the words users say to invoke the flash briefing or news request (*utterances*) and the format of the content so that Alexa can provide it to the customer. |
| A skill that enables users to select, listen to, and control audio content streamed through an Alexa-enabled device. | [Music Skill API](https://developer.amazon.com/en-US/docs/alexa/music-skills/understand-the-music-skill-api.html)  When you build a music skill, the voice interaction model is defined and handled for you. Alexa interprets user *utterances* and sends messages to your skill that communicate these requests. |

For more details about the differences between different types of skills, see [Understand the Different Skill Models](https://developer.amazon.com/docs/ask-overviews/understanding-the-different-types-of-skills.html).

What Do I Build?

You create a cloud-based service that handles the requests for the skill type and host it in the cloud. The Alexa service routes incoming requests to the appropriate service.

Different types of skills require different types of services:

For a *custom skill*, you code either an [AWS Lambda function](http://aws.amazon.com/lambda/) or a *web service*:

* AWS Lambda (an [Amazon Web Services](http://aws.amazon.com/) offering) is a service that lets you run code in the cloud without managing servers. Alexa sends your code user requests and your code can inspect the request, take any necessary actions (such as looking up information online) and then send back a response. You can write Lambda functions in Node.js, Java, Python, C#, or Go.
* Alternatively, you can write a *web service* and host it with any cloud hosting provider. The web service must accept requests over HTTPS. In this case, Alexa sends requests to your web service and your service takes any necessary actions and sends back a response. You can write your web service in any language.
* Regardless of how you create your service, you also create a custom *interaction model* for the skill. This defines the *requests* the skill can handle and the *words users can say* to invoke those requests.

For a skill that controls smart home devices such as lights, thermostats, and entertainment devices you can use the Smart Home Skill API. In this case, you develop an AWS Lambda function that accepts device directives from Alexa:

* You provide code to handle directives in an AWS Lambda function.
* Your skill receives requests in the form of *device directives* to control a particular device. Your code then handles the request appropriately (for example, by turning on the requested light or turning up the volume).
* All voice interactions with the user are handled by the Smart Home Skill API. You don't need to define the words users say to use the skill.

For a skill that controls video content you can use the Video Skill API. In this case, you develop a lambda function that accepts device directives from Alexa:

* You provide code to handle directives in an AWS Lambda function.
* Your skill receives requests in the form of *device directives* to control a video service. Your code then handles the request appropriately (for example, by playing a movie).
* All voice interactions with the user are handled by the Video Skill API. You don't need to define the words users say to use the skill.

For a skill that provides content such as news, lists, or comedy for a customer's flash briefing, you can use the Flash Briefing Skill API. In this case, you create the skill in the developer console and configure one or more JSON or RSS feeds that contain the content:

* To receive your content as a part of their flash briefing, a customer enables your flash briefing skill in the Alexa app, and turns on one or more content feeds.
* All voice interactions with the user are handled by the Flash Briefing Skill API. You don't need to define the words users say to use the skill.
* You supply one or more reliable content feeds in RSS or JSON format. The content can be audio content that Alexa plays to the customer, or text content that Alexa reads to the customer. You should own the content or have the rights to distribute it.

Understand the Different Skill Models

The first step in building a new skill is to decide *what your skill will do*. This determines how your skill integrates with the Alexa service and what you need to build. The Alexa Skills Kit supports building several different types of skills.

* [Skill Models](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#skill-models)
* [Custom Interaction Model](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#custom-interaction-model)
* [Smart Home Skills (Pre-built Model)](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#smart-home-skills-pre-built-model)
* [Flash Briefing Skills (Pre-built Model)](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#flash-briefing-skills-pre-built-model)
* [Video Skills (Pre-built Model)](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#video-skills-pre-built-model)
* [Music Skills (Pre-built Model)](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#music-skills-pre-built-model)
* [List Skills](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#list-skills)
* [Next: Understand How Users Interact with Skills](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#next-understand-how-users-interact-with-skills)
* [Jump In](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#jump-in)

Skill Models

Every skill has an *interaction model* that determines the requests the skill can handle and the words users say to invoke those requests. You can define this yourself with a *custom model*. The Alexa Skills Kit also provides *pre-built models* in which the possible requests and utterances are pre-defined for you.

Custom Interaction Model

For the most control over the user's experience, build a skill with a *custom interaction model*. This is a *custom skill*.

For a custom skill, *you* (as the developer) define:

* The requests the skill can handle. These are defined as *intents*. For example, a skill could do any of the following:
  + Look up tide information
  + Order a pizza
  + Request a taxi
  + Engage the user in a game, such as word puzzles or trivia
  + Just about any other action you can imagine!
* The words users say to make (or invoke) those requests. This is the *interaction model*, and it provides the *voice user interface* by which users communicate with the skill. Continuing the above examples:
  + "Get high tide for Seattle" (this phrase would be mapped to a TideRequest intent).
  + "Order a large pepperoni pizza" (this phrase would be mapped to an OrderPizza intent).
  + "Order a car" (this phrase would be mapped to an OrderCar intent).
* The visual and touch interactions that users will experience and can invoke. Alexa-enabled devices with a screen support visual displays and touch interactions, so you can create a skill that uses a combination of voice, visual, and touch interactions, or you can opt to have a skill that does not support any screen functionality.
* The name Alexa uses to identify your skill, called the *invocation name*. Users include this when making a request. For example, the skill for looking up tides could be called "tide pooler".

Putting this all together, a user could say this:

**User:** *get high tide for Seattle* from **Tide Pooler**

Alexa understands this request and sends the TideRequest intent to the service for the Tide Pooler skill.

A custom skill can handle any kind of request, so long as you can create the code to fulfill the request and provide appropriate data in the interaction model to let users invoke the request. This is the most flexible kind of skill you can build, but also the most complex, since you need to provide the interaction model.

Smart Home Skills (Pre-built Model)

For building a skill to control smart home devices such as cameras, lights, locks, thermostats, and smart TVs, you should use the *Smart Home pre-built model*. This gives you less control over the user's experience, but simplifies development since you don't need to create the voice user interface yourself. These skills are also easier for end users to invoke, since they don't need to remember any invocation name and can make requests such as "Alexa, turn on the living room lights."

For this type of skill, the *Smart Home Skill API* defines:

* The requests the skill can handle. These requests are called *device directives*. Examples include:
  + turn on / turn off
  + increase / decrease the temperature
  + change the dimness or brightness for a light
  + lock a door
  + change the channel on a television
  + view a live video stream from a smart home camera on Echo Show or Fire TV.
* The words users say to make (or invoke) those requests. For example:
  + "turn off the living room lights"
  + "increase the temperature by two degrees"
  + "dim the living room lights to 20%"
  + "lock the back door"
  + "change channel to PBS"
  + "show the front door camera"

*You* (as the developer) define how your skill responds to a particular directive. For instance, you write the code that makes a light turn on when your skill receives a "turn on the light" directive. This code is hosted as an AWS Lambda function. Note that a skill built with the Smart Home Skill API can respond only to the requests (device directives) supported by the API.

Flash Briefing Skills (Pre-built Model)

A flash briefing skill is the only way that you can provide content for a customer's flash briefing.

For this type of skill, the Flash Briefing Skill API defines:

* The words users say to make (or invoke) those requests. For example:
  + "give me my flash briefing"
  + "tell me the news"

You (as the creator) define:

* The name, description and images for a flash briefing skill. This helps a customer choose your skill in the skill store.
* One or more content feeds for a flash briefing skill. These feeds can contain audio content that is played to the customer or text content that Alexa reads to the customer.

Video Skills (Pre-built Model)

A video skill enables you to provide video content such as TV shows and movies for customers.

For this type of skill, the Video Skill API defines:

* The words users say to make (or invoke) those requests. For example:
  + "play Manchester by the Sea"
  + "change to channel 4"

You (as the creator) define:

* The name, description and images for a video skill. This helps a customer choose your skill in the skill store.
* The requests the skill can handle such as playing and searching for video content and how video content search results display.

Music Skills (Pre-built Model)

A music skill enables you to provide audio content such as songs, playlists, or radio stations for Alexa users. For this type of skill, the Music Skill API handles the words (*utterances*) a user can say to request and control audio content, and turns these utterances into requests that are sent to your skill. Your skill code (AWS Lambda function) handles these requests and responds appropriately, sending back audio content for the user to listen to on an Alexa-enabled device.

List Skills

A list skill facilitates the use of [list events](https://developer.amazon.com/docs/smapi/list-events-in-alexa-skills.html) in the skill service. Thus, the skill can understand and react to changes that happen to lists from top-level utterances on Alexa.

A [list skill is created](https://developer.amazon.com/docs/smapi/steps-to-create-a-list-skill.html) through the [ASK CLI (Alexa Skills Kit Command Line Interface)](https://developer.amazon.com/docs/smapi/ask-cli-intro.html).

A list skill may include an optional custom component, which can include any features allowed for [custom skills](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-the-different-types-of-skills.html#custom-skills), in addition to list skill features.

In respect to the list component of a list skill, the List API defines:

The intents and user actions that will trigger list events, such as:

* Adding an item to a list
* Removing an item from a list
* Updating an item on a list

You (as the creator) define:

* The endpoint to receive list events
* Handlers to process and respond to list events

# Index of Skill Types

The Alexa Skills Kit (ASK) categorizes skills into types based on their capabilities or features. The following table lists the available skill types alphabetically. Follow the link to each skill type for full details.

For skills with a pre-built voice model, users interact with Alexa by using a set of pre-defined utterances. For skills with a custom voice model, you define the utterances that you want users to say to Alexa.

* **Note:** Not all skill types are available in all locales. For more information, see [List of Capability Interfaces and Supported Locales](https://developer.amazon.com/en-US/docs/alexa/device-apis/list-of-interfaces.html).

| Skill Type | Voice Model | Description | Common APIs and Other Reference |
| --- | --- | --- | --- |
| [Automotive](https://developer.amazon.com/en-US/docs/alexa/custom-skills/best-practices-for-designing-alexa-skills-for-automotive.html) | Custom | Build skills adapted to the automotive environment. | — |
| [Baby activity](https://developer.amazon.com/en-US/docs/alexa/health/overview.html) | Pre-built | Build baby activity skills for your health and wellness apps and devices. | [Health.Sleep](https://developer.amazon.com/en-US/docs/alexa/health/alexa-health-sleep.html), [Health.Weight](https://developer.amazon.com/en-US/docs/alexa/health/alexa-health-weight.html) |
| [Conferencing](https://developer.amazon.com/en-US/docs/alexa/alexa-for-business/build-skills-for-conferencing-devices.html) | Pre-built | Build skills that give customers voice access to their business calendars and meetings. | [Calendar](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-calendar.html), [MeetingClientController](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-meetingclientcontroller.html) |
| [Cooking](https://developer.amazon.com/en-US/docs/alexa/smarthome/build-smart-home-skills-for-cooking-appliances.html) | Pre-built | Build skills for cooking appliances. | [TimeController](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-cooking-timecontroller.html) |
| [Custom](https://developer.amazon.com/en-US/docs/alexa/custom-skills/understanding-custom-skills.html) | Custom | Build a skill with a custom voice interaction model. | [Requests and Responses](https://developer.amazon.com/en-US/docs/alexa/custom-skills/request-and-response-json-reference.html), [Request Types](https://developer.amazon.com/en-US/docs/alexa/custom-skills/request-types-reference.html) |
| [Echo button](https://developer.amazon.com/en-US/docs/alexa/echo-button-skills/understand-echo-button-skills.html) | Custom | Build skills that use [Echo Buttons](https://www.amazon.com/Echo-Buttons-Alexa-Gadget-Pack/dp/B072C4KCQH) to display lights and capture input from users quickly, such as games, puzzles, and so on. | [GameEngine](https://developer.amazon.com/en-US/docs/alexa/echo-button-skills/gameengine-interface-reference.html), [GadgetController](https://developer.amazon.com/en-US/docs/alexa/echo-button-skills/gadgetcontroller-interface-reference.html) |
| [Flash briefing](https://developer.amazon.com/en-US/docs/alexa/flashbriefing/understand-the-flash-briefing-skill-api.html) | Pre-built | Build a flash briefing skill to provide Alexa customers with news headlines and other short content. | — |
| [List](https://developer.amazon.com/en-US/docs/alexa/custom-skills/access-the-alexa-shopping-and-to-do-lists.html) | Custom | Build skills that read and update a user's Alexa lists. | — |
| [Meetings](https://developer.amazon.com/en-US/docs/alexa/alexa-for-business/understand-meetings-skills.html) | Pre-built | Build a skill to search for and reserve meeting rooms. | [Reservation.Room](https://developer.amazon.com/en-US/docs/alexa/alexa-for-business/alexa-business-reservation-room.html) |
| [Music](https://developer.amazon.com/en-US/docs/alexa/music-skills/understand-the-music-skill-api.html) | Pre-built | Build skills that control audio content streamed through Alexa-enabled devices. | Audio.PlayQueue, Media.Search |
| [Networking](https://developer.amazon.com/en-US/docs/alexa/networking/overview.html) | Pre-built | Build a skill to model a home Wi-Fi network and the devices connected to it. Enable and disable internet and network access for individual devices. | [ConnectedDevice](https://developer.amazon.com/en-US/docs/alexa/networking/alexa-networking-connecteddevice.html) |
| [Smart home](https://developer.amazon.com/en-US/docs/alexa/smarthome/understand-the-smart-home-skill-api.html) | Pre-built | Build skills that enable users to voice-control their smart home devices. | [LockController](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-lockcontroller.html), [PowerController](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-powercontroller.html), [ThermostatController](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-thermostatcontroller.html) |
| [Video](https://developer.amazon.com/en-US/docs/alexa/video/understand-the-video-skill-api.html) | Pre-built | Build skills that enable users to control video devices and consume video content. | [KeypadController](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-keypadcontroller.html), [VideoRecorder](https://developer.amazon.com/en-US/docs/alexa/device-apis/alexa-videorecorder.html) |

Understand How Users Interact with Skills

When a user speaks to a device with Alexa, the speech is streamed to the Alexa service in the cloud. Alexa recognizes the speech, determines what the user wants, and then sends a structured request to the particular skill that can fulfill the user's request. All speech recognition and conversion is handled by Alexa in the cloud.

Every Alexa skill has an *interaction model* defining the words and phrases users can say to make the skill do what they want. This model determines how Alexa communicates with your users.

The following sections provide examples and more detail around how users communicate with Alexa and what you need to do as a developer when designing a skill.

Note that the interaction model described here is focused on voice interaction. If you are developing a skill that includes screen display, video, or touch interaction on an Alexa-enabled device with a screen, also refer to [Best Practices for Designing Skills for Alexa-Enabled Devices With a Screen](https://developer.amazon.com/docs/custom-skills/best-practices-for-designing-skills-for-alexa-enabled-devices-with-a-screen.html).

* [How Users Interact with Alexa](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#how-users-interact-with-alexa)
* [What is an Interaction Model?](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#what-is-an-interaction-model)
* [Examples of Interaction Models](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#examples-of-interaction-models)
  + [Interact with a Custom Skill](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#interact-with-a-custom-skill)
  + [Interact with the Smart Home Skill API](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#interact-with-the-smart-home-skill-api)
  + [Interact with the Video Skill API](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#interact-with-the-video-skill-api)
  + [Interact with the Flash Briefing Skill API](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#interact-with-the-flash-briefing-skill-api)
  + [Interact with the Music Skill API](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#interact-with-the-music-skill-api)
* [Next: Learn what you need to get started](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#next-learn-what-you-need-to-get-started)
* [Jump In](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html#jump-in)

How Users Interact with Alexa

End users interact with Alexa's abilities in the same way – by waking the device with the wake word (or a button for a device such as the Amazon Tap) and asking a question or making a request.

In addition, with Alexa-enabled devices with a screen, the end user can touch the screen to interact with Alexa, if the skill supports this interaction.

For example, users interact with the built-in Weather service like this:

**User:** Alexa, what's the weather?  
**Alexa:** Right now in Seattle, there are cloudy skies…

Interacting with a custom-built skill to look up tide information is very similar, although the user can include a name identifying the skill ("tide pooler" in this example):

**User:** Alexa, get high tide for Seattle from Tide Pooler  
**Tide Pooler:** Wednesday February 17th in Seattle, the first high tide will be around 1:42 in the morning, and will peak at about 10 feet…

Similarly, a user can tell a skill to control a particular cloud-enabled device such as a light. In this case, the light must be connected to the Internet (such as via a smart home hub). The user names the specific light ("living room lights" in this example):

**User:** Alexa, turn on the living room lights  
*A light previously configured and named* ***living room lights*** *is turned on.*  
**Alexa:** OK.

What is an Interaction Model?

In the context of Alexa, an interaction model is somewhat analogous to a graphical user interface in a traditional app. Instead of clicking buttons and selecting options from dialog boxes, users make their requests and respond to questions by voice:

| Action | Voice User Interface (Interaction Model) | Typical Graphical User Interface |
| --- | --- | --- |
| Make a request | User says, *"Alexa, get high tide from tide pooler."* | User clicks a button. |
| Collect more information from the user | Alexa replies, *"For what city?"* and then waits for a response. | App displays a dialog box and waits for user to select an option. |
| Provide needed information | User replies, *"Seattle."* | User selects options and chooses **OK**. |
| User's request is completed | Alexa speaks the requested information:  *"Wednesday February 17th in Seattle, the first high tide will be…"*" | App displays the results of the request. |

When users speak questions and make requests, Alexa uses the interaction model to interpret and translate the words into a *specific request* that can be handled by a *particular skill*. The request is then sent to the skill.

You define your own interaction model when creating a custom skill. The Smart Home Skill API, Video Skill API, Music Skill API, and others provide a built-in interaction model.

Examples of Interaction Models

Interact with a Custom Skill

Note this phrase a user can speak:

**User:** Alexa, *get high tide for Seattle* from *Tide Pooler*.

* "*Tide Pooler*" is the *invocation name* that identifies a particular skill.
* "*get high tide for Seattle*" is a phrase in Tide Pooler's interaction model. This phrase is mapped to a specific intent supported by this skill.

Alexa uses this custom interaction model to create a structured representation of the request called an *intent*. Alexa sends the intent to the Tide Pooler skill. The skill can then look up tide information and send back a response.

Users can also invoke custom skills without using an invocation name: Alexa, get the high tide for Seattle. In this case, Alexa attempts to find a skill that can respond to the user's utterance. Alexa then creates the *intent* and sends it to the skill.

Interact with the Smart Home Skill API

Note this phrase a user can speak:

**User:** Alexa, turn on the **living room lights**

* "*turn on the…*" is a phrase recognized by Alexa's built-in interaction model. Alexa recognizes that this is a request to turn on a light.
* The words "*living room lights*" identify a particular device that the user has previously configured and named. Note that this is the name of the device to control, *not* the name of the skill. The user does not need to say an invocation name for this type of skill.

Alexa uses the pre-built interaction model for smart home requests to create a structured representation of the request, called a *device directive*. Alexa sends the device directive to the specific skill that can control the "living room lights" device. This skill turns on the specified lights by communicating with the device cloud over the Internet, then returns a response indicating whether it was successful.

Interact with the Video Skill API

Note this phrase a user can speak:

**User:** Alexa, play **Manchester by the Sea**

* "*play…*" is a phrase recognized by Alexa's built-in interaction model. Alexa recognizes that this is a request to play content.
* The words "*Manchester by the Sea*" identify a particular video title. Note that the user does not need to say an invocation name for this type of skill.

Alexa uses the pre-built interaction model for video skills to create a structured representation of the request, called a *device directive*. Alexa sends the device directive to the specific skill that controls playback of video content. This skill plays the content by communicating with the service that controls the content, and returns a response indicating whether it was successful.

Interact with the Flash Briefing Skill API

Note this phrase a user can speak:

**User:** Alexa, what's my flash briefing?

"*What's my flash briefing*" is a phrase recognized by Alexa's built-in interaction model. Alexa recognizes that this is a request to read and stream content from feeds that the user previously selected.

Alexa uses the pre-built interaction model for content requests to invoke the flash briefing feature. Alexa loads the content feeds the user has selected to include in their flash briefing and streams the content.

Interact with the Music Skill API

Note this phrase a user can speak:

**User:** Alexa, play Poker Face by Lady Gaga on *skill name*.

* "*play…*" is a phrase recognized by Alexa's built-in interaction model. Alexa recognizes that this is a request to play content.
* "*Poker Face by Lady Gaga*" identifies a particular song title.
* "*skill name*" identifies the skill to invoke to play this content.

Alexa uses the pre-built interaction model for music skills to create a structured representation of the request, then sends that request to the specified skill. This skill communicates with the service that manages the content, then returns a response with the requested content for playback on an Alexa-enabled device.

Requirements to Build a Skill

Once you know what type of skill you want to build, you can determine what you need to get started.

* [Create an Amazon developer account](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/requirements-to-build-a-skill.html#create-an-amazon-developer-account)
* [What do you need for a custom skill?](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/requirements-to-build-a-skill.html#what-do-you-need-for-a-custom-skill)
* [What do you need for a smart home skill?](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/requirements-to-build-a-skill.html#what-do-you-need-for-a-smart-home-skill)
* [What do you need for a video skill?](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/requirements-to-build-a-skill.html#what-do-you-need-for-a-video-skill)
* [What do you need for a flash briefing skill?](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/requirements-to-build-a-skill.html#what-do-you-need-for-a-flash-briefing-skill)
* [What do you need for a music skill?](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/requirements-to-build-a-skill.html#what-do-you-need-for-a-music-skill)
* [Next Steps](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/requirements-to-build-a-skill.html#next-steps)

Create an Amazon developer account

To build any type of skill, you need an Amazon developer account. Go to the developer console at <https://developer.amazon.com/>. You can use an existing Amazon account to sign in, or you can create a new Amazon account.

Follow the prompts to enter your registration information and consent to the Amazon Developer Services Agreement. Once you have done this, you can build skills for Alexa using this account.

To build a skill, you must first create your skill configuration. Your skill configuration collects information about the skill, such as its name, the type of interaction model to use, the endpoint or content feed, and other information. The Alexa service uses the configuration to determine which user requests should be sent to the service for your skill.

You can create your skill configuration in the developer console, or you can use the [Alexa Skills Kit Command Line Interface](https://developer.amazon.com/en-US/docs/alexa/smapi/ask-cli-intro.html) or the [Skill Management API](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/smapi/smapi-overview.html).

What do you need for a custom skill?

To build a custom skill, you need:

* An Internet-accessible endpoint for hosting your cloud-based service.

The simplest option for this is to use [AWS Lambda](http://aws.amazon.com/lambda/) (an [Amazon Web Services](http://aws.amazon.com/) offering). In this case, you need an account with Amazon Web Services in addition to your Amazon developer account.

Alternatively, you can build and host an HTTPS web service. In this case, you will need a cloud hosting provider and an SSL certificate.

* A development environment appropriate for the programming language you plan to use. You can author a Lambda function in Node.js, Java, Python, C#, or Go. You can author a web service in any language appropriate for web services.
* A publicly accessible web site to host any images, audio files, or video files that you use in your skill. If you have no such files other than for a skill icon, you do not need to host any resources. One possible solution is to use an Amazon Simple Storage Service (S3) instance, (an [Amazon Web Services](http://aws.amazon.com/) offering).
* Optionally, a device with Alexa for testing. Skills work with all devices with Alexa, such as the Amazon Echo, Amazon Echo Dot, Amazon Tap, Fire TV, and devices that use the Alexa Voice Service.

If you don't have a device, you can use a simulator on the [**Test** page](https://developer.amazon.com/en-US/docs/alexa/devconsole/test-your-skill.html) for testing. You can also see what the display templates for Echo Show and Echo Spot look like, although the display is not interactive. In addition, if your skill includes display and touch interactions, you require an Alexa-enabled device with a screen to test it.

See [Understanding Custom Skills](https://developer.amazon.com/docs/custom-skills/understanding-custom-skills.html).

What do you need for a smart home skill?

To build a smart home skill, you need:

* The cloud-enabled device that you want to control (such as a light, switch, or thermostat). The device must be controllable via a cloud-based service that you have access to.
* An account with Amazon Web Services, needed for a Lambda function to host the skill code. Skills built with the Smart Home API must be hosted on [AWS Lambda](http://aws.amazon.com/lambda/).
* A development environment appropriate for the programming language you plan to use. You can author a Lambda function in Node.js, Java, Python, C#, or Go.
* An Alexa-enabled device for testing. Skills work with all Alexa-enabled devices, such as the Amazon Echo, Amazon Echo Dot, Echo Show, Fire TV Cube, Fire TV, and devices that use the Alexa Voice Service.

If you don't have a device, you can use a simulator on the [**Test** page](https://developer.amazon.com/en-US/docs/alexa/devconsole/test-your-skill.html) for testing.

See [Understanding the Smart Home Skill API](https://developer.amazon.com/docs/smarthome/understand-the-smart-home-skill-api.html).

What do you need for a video skill?

To build a video skill, you need:

* A cloud-enabled video service provider. You must have access to the cloud-based service or it must have a public API.
* An account with Amazon Web Services, needed for a Lambda function to host the skill code. Skills built with the Video Skill API must be hosted on [AWS Lambda](http://aws.amazon.com/lambda/).
* A development environment appropriate for the programming language you plan to use. You can author a Lambda function in Node.js, Java, Python, C#, or Go.
* An Alexa-enabled device for testing.

See [Understanding the Video Skill API](https://developer.amazon.com/docs/video/understand-the-video-skill-api.html).

What do you need for a flash briefing skill?

To build a flash briefing skill, you need:

* An Internet-accessible *content feed* (using RSS or JSON) that refreshes with new content on an ongoing basis.

See [Understanding the Flash Briefing Skill API](https://developer.amazon.com/docs/flashbriefing/understand-the-flash-briefing-skill-api.html).

What do you need for a music skill?

To create a music skill, you need the following:

* An [Amazon developer account](https://developer.amazon.com/). Sign up is free.
* An Amazon Alexa-enabled device, such as Amazon Echo, registered to your Amazon developer account.
* A streaming music service with a cloud API to control it.
* The ability to provide your music catalog metadata to Amazon on a regular basis (for example, weekly) for voice modeling and entity resolution purposes.
* Permission to stream the content that your music skill or music service makes available to users.
* An [AWS account](https://aws.amazon.com/). You host your skill code as an AWS Lambda function.
* Knowledge of one of the programming languages supported by AWS Lambda: Node.js, Java, Python, C#, or Go.
* A basic understanding of [OAuth 2.0](http://oauth.net/2/) if your skill uses [account linking](https://developer.amazon.com/en-US/docs/alexa/account-linking/understand-account-linking.html).

Alexa Skills Kit Glossary

This document defines common terms used throughout the Alexa Skills Kit documentation.

[A](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#a)| [C](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#c)| [D](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#d)| [E](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#e)| [F](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#f)| [H](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#h)| [I](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#i)| [L](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#l)| [M](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#m)| [N](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#n)| [O](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#o)| [P](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#p)| [R](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#r)| [S](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#s)| [T](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#t)| [U](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#u)| [V](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#v)| [W](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#w)|

A

access token

A credential that represents the end user (resource owner) in another system. A token should identify the user in the other system. The access token is included in the requests sent to your skill if the user has successfully linked their accounts.

access token URI (access token endpoint)

The URI (Uniform Resource Identifier) of the access token endpoint for the authentication server. The Alexa service calls this endpoint to exchange the [authorization code](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#authorization-code) for the [access token](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#access-token). For an Alexa skill, you specify this endpoint in the developer console, in the Access Token URI field on the [Build > Account Linking](https://developer.amazon.com/en-US/docs/alexa/devconsole/build-your-skill#account-linking) page.

account linking

An Alexa Skills Kit feature that lets you connect the identity of the end user with an account in another system. For example, a Car Hailer custom skill for ordering a ride needs to access the Car Hailer service as a specific user. Similarly, a smart home skill for controlling a light needs to connect the Alexa user with an account in the device cloud. See [Understand Account Linking](https://developer.amazon.com/en-US/docs/alexa/account-linking/understand-account-linking.html).

Alexa app

The companion app for Alexa customers to set up devices, change settings, and see the displayed output from interactions with Alexa.

Alexa-enabled device

A device that provides access to the Alexa service. Examples include Amazon Echo, Amazon Echo Dot, Amazon Tap, Echo Show, and devices made by other manufacturers that use the Alexa Voice Service.

Alexa Presentation Language (APL)

A composable, responsive layout language from Amazon that is used to create interactive visual experiences for Alexa. See [Understand Alexa Presentation Language (APL)](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/understand-apl.html).

Alexa service / Alexa

The cloud-based voice service that powers [Alexa-enabled devices](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#alexa-enabled-device) made by Amazon or other manufacturers. You can give Alexa new abilities by creating your own cloud-based service that accepts requests from Alexa and returns responses.

Alexa skill

See [Skill](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#skill).

Alexa Skills Kit

A collection of APIs, tools, and documentation for giving Alexa new capabilities. See [Build Skills with the Alexa Skills Kit](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/build-skills-with-the-alexa-skills-kit.html).

Amazon Polly

A service that turns text into lifelike speech, allowing you to create applications that talk, and build entirely new categories of speech-enabled products. The text-to-speech service uses advanced deep learning technologies to synthesize speech that sounds like a human voice. With dozens of lifelike voices across a variety of languages, you can select the ideal voice and build speech-enabled applications that work in many different countries. For more information, see [Amazon Polly](https://aws.amazon.com/polly/).

annotation

A test utterance annotated with the intent and slot values you expect the utterance to trigger. You can create an annotation set containing a group of annotations, then run an evaluation with the [NLU evaluation tool](https://developer.amazon.com/en-US/docs/alexa/custom-skills/batch-test-your-nlu-model.html) to batch test the accuracy of your NLU model.

ask

One of the words a user can say to ask Alexa to invoke a particular custom skill. This is used in combination with the [invocation name](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#invocation-name) for the skill. For example, Alexa, ask Astrology Daily for my horoscope. There are several phrases users can say to start a conversation with Alexa. See [Understanding How Users Invoke Custom Skills](https://developer.amazon.com/en-US/docs/alexa/custom-skills/understanding-how-users-invoke-custom-skills.html).

authorization code

A value used by [authorization code grant](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#auth-code-grant) to allow a client to request an access token from an authorization server. Used with authorization code grant.

authorization code grant

A grant type for requesting the end user's authorization to access their information in another system. In authorization code grant, the authorization server returns a code once the user logs in. Alexa then uses this code to request an access token / refresh token pair from the server, using the [access token endpoint (URI)](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#access-token-uri). The refresh token can then be used to request a new access token after the old token expires.

authorization server

The server that authenticates the identity of the resource owner and issues access tokens. This can be the same as the [Resource server](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#resource-server), but it does not have to be. One of the four roles defined by OAuth 2.0, used in account linking. For example: if your skill connects to a service for ordering rides (Car Hailer), the authorization server authenticates the user and provides an access token that can be used to get the user's Car Hailer profile from the resource server.  
  
You specify this in the developer console, in the Authorization URI field on the [Build > Account Linking](https://developer.amazon.com/en-US/docs/alexa/devconsole/build-your-skill#account-linking) page.

authorization URI

The URI (Uniform Resource Identifier) of the page for entering login credentials for a service. The user is redirected to the authorization URI when starting the account linking process for a skill. Once the user has authenticated, this page redirects the user back to an Alexa-specified redirect\_uri.

automatic speech recognition (ASR)

A technology that converts spoken words into text. ASR is the first step in enabling technologies like Alexa to respond when you ask, Alexa, what's it like outside?

AWS Lambda

An Amazon Web Services (AWS) compute service that runs your code in response to events and automatically manages the compute resources for you. This lets you run code (referred to as a [Lambda function](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#aws-lambda-function)) in the cloud without managing servers. The code for your skill must be hosted as a Lambda function and is required for smart home skills. You can also choose to use a Lambda function for the service for a custom skill. [AWS Lambda](https://aws.amazon.com/lambda/) is a service offering by [Amazon Web Services](https://aws.amazon.com/).

AWS Lambda function

The code uploaded to AWS Lambda. Lambda supports coding in Node.js, Java, Python, or C#. A smart home skill must be implemented as a Lambda function. You can also choose to use a Lambda function for the service for a custom skill.

C

card

See [Home card](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#home-card) or [Detail card](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#detail-card).

client

The application making the requests to the resource server on behalf of the resource owner, with the resource owner's authorization. In the context of account linking for skills, this role is shared by the Alexa service and the skill. Alexa handles getting the resource owner's authorization and the skill uses the access token to access the protected resource in the resource server. One of the four roles defined by OAuth 2.0, used in account linking.

cloud-based service

See [Service](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#service).

companion app

See [Alexa app](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#alexa-app).

component

In Alexa Presentation Language (APL), a primitive UI element which can be displayed on the viewport. APL provides a set of components for UI elements such as text, image, list and video. You can combine components to create custom layouts. The Alexa Design System for APL also provides a set responsive components that can be used like components, such as a Button component that combines the primitive components Text, TouchWrapper, and Frame and builds in responsiveness to different viewports.

confirmation

When Alexa says something to make sure the customer knows she understood them correctly. Types of confirmation include [Implicit confirmation](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#implicit-confirmation) and [Explicit confirmation](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#explicit-confirmation).

consumable purchase

An [in-skill purchase](https://developer.amazon.com/en-US/docs/alexa/in-skill-purchase/isp-overview.html) that provides content or features that can be purchased, depleted, and purchased again. For example, hints for a game, in-game currency or extra lives.

conversation

See [Interaction](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#interaction).

custom interaction model

The interaction model for a custom skill. A custom interaction model consists of an intent schema that defines the requests the skill can handle and a set of sample utterances that customers can say to invoke those requests.

custom skill

A skill that uses a custom interaction model. You define the requests your skill can handle (intents) and the words customers say to make (or invoke) those requests (sample utterances). The mapping between the intents and sample utterances creates the interaction model or voice user interface for the skill.

D

data-binding

In Alexa Presentation Language (APL), the method used to retrieve data from a data source supplied by you (the developer) or device information and display it using APL components. Data binding lets you separate your presentation logic (the APL document) from your source data. You use data binding expressions to bind data to APL components or layouts.

data-binding context

The collection of data sources and data available to an APL document for data binding. The data is structured in key-value pairs and includes the data source that accompanies the APL document, information about the device viewport (such as viewport.width), device capabilities, and environment information (such as environment.aplVersion). The context can also contain derived data. For example, data from a specific element in a array when a data source is bound to a component such as Sequence that can iterate over a collection and inflate each element of the array.

data-binding expression

In Alexa Presentation Language (APL), an expression that is dynamically evaluated when Alexa displays your document. You use data-binding expressions to bind component properties to your data source, and to write conditional logic to hide and show components depending on the viewport characteristics. Data-binding expressions begin with a dollar sign followed by curly brackets: ${..}. See [APL Data-Binding Evaluation](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-data-binding-evaluation.html) and [APL Data-Binding Syntax](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-data-binding-syntax.html).

data-binding syntax

In Alexa Presentation Language (APL), the syntax used to write a data binding expression. See [APL Data-Binding Syntax](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-data-binding-syntax.html).

delegate

Lets Alexa determine and complete the next step during a dialog between the user and the skill. Alexa uses the skill's dialog model to determine the next step in the dialog and ask the user for the information. You can [delegate the dialog manually or automatically](https://developer.amazon.com/en-US/docs/alexa/custom-skills/delegate-dialog-to-alexa.html).

detail card

A card displayed in the Alexa app with information about the skill and how to use it. Customers can review detail cards and enable the skills they want. You enter most of the information displayed on the detail card on the [Launch Your Skill](https://developer.amazon.com/en-US/docs/alexa/devconsole/launch-your-skill.html) page.

developer console

A development tool that provides a streamlined experience to help you create, manage, and publish Alexa skills.

device cloud

Back-end cloud service that can control a cloud-enabled device. For a smart home skill, you write code hosted as a Lambda function that translates commands from the Alexa smart home skill API to the device cloud.

device cloud account

Unique customer account used to access the device cloud. The customer links the device cloud account with the Alexa service using the Alexa app. OAuth 2.0 is the required mechanism for account linking.

device directive

A set of data and instructions, expressed in JSON, sent from Alexa to a smart home or video skill.

device discovery

The process by which the Alexa smart home skill API or Video Skill API discovers the devices that can be controlled with a skill.

device event

A response to a device directive, expressed in JSON, sent from a smart home or video skill to Alexa.

device with Alexa

A device that provides access to the Alexa service. Examples include Amazon Echo, Amazon Echo Dot, Amazon Tap, and third-party devices that use the Alexa Voice Service.

dialog errors

When something unexpected happened in the conversation between Alexa and the customer. Types of dialogue errors include [low confidence errors](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#low-confidence-error), [timeouts/silence/no input](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#timeout-error), and [false accepts](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#false-accept-error). For recommendations on how to handle dialogue errors, see the section on [Error messages](https://developer.amazon.com/docs/alexa-design/adaptable.html#handle-errors-gracefully) in the Alexa Design Guide.

dialog model

A structure that identifies the prompts and user utterances to collect, validate, and confirm slot values and intents. You use a dialog model to let Alexa determine the next step in a conversation and ask the user for more information. A dialog model also improves accuracy when you manage the dialog manually with the [Dialog](https://developer.amazon.com/en-US/docs/alexa/custom-skills/dialog-interface-reference.html) directives such as Dialog.ElicitSlot. To create a dialog model, see [Define the Dialog to Collect and Confirm Required Information](https://developer.amazon.com/en-US/docs/alexa/custom-skills/define-the-dialog-to-collect-and-confirm-required-information.html).

directive

See [device directive](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#device-directive).

directive language

JSON protocol that enables communication between the Alexa smart home skill API and a smart home skill.

display template

A template that is used by Alexa-enabled devices with a screen to display a combination of text and images in a specified format. There are several body templates, which are used to display content, and several list templates, which allow selection of content. Display templates are displayed as cards in the Alexa app if the user does not have an Alexa-enabled device with a screen.

E

Echo app

See [Alexa app](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#alexa-app).

endpoint

Represents a physical device, virtual device, group or cluster of devices or a software component that is primarily used in smart home and video skills. An endpoint identifies the target for a directive and the origin of an event from a skill.

error message

The message delivered to a customer when an utterance or technical error occurs during a dialog.

evaluation (NLU)

A test that evaluates the natural langauge understanding (NLU) model built from your skill's interaction model, using a specified annotation set. This can be especially useful for regression testing as you iterate on your interaction model. See [Batch Test Your Natural Language Understanding (NLU) Model](https://developer.amazon.com/en-US/docs/alexa/custom-skills/batch-test-your-nlu-model.html).

example phrase

A phrase showing customers what they need to say to begin using your custom skill. You'll enter these phrases in the Distribution tab of the developer console. The phrases must also be included in your list of [Sample utterances](https://developer.amazon.com/en-US/docs/alexa/devconsole/launch-your-skill.html#example-phrases-for-custom-skills).

exit command

When the customer says a command like exit or stop to end the interaction.

exit message

The message delivered when a customer asks the skill to stop/exit or when the conversation with the skill comes to a natural end.

explicit confirmation

A prompt that repeats back what Alexa heard and explicitly asks the customer to confirm whether they were correct. For example, Alexa, ask Astrology Daily for my horoscope. Alexa would respond with, You wanted a horoscope from Astrology Daily, right?

F

false accept error

When Alexa has mid to high confidence that she correctly understood what the customer said, but she actually misunderstood.

full intent invocation

A customer's request that contains all information Alexa needs to make the request actionable. For example, Alexa, ask History Buff what happened on June 3rd.

H

help message

The message delivered to a customer who either asks the skill for help or reaches an error message too many consecutive times. A master help message delivers high-level information about the skill and its features that includes similar information as the first-time welcome message. Contextual help messages are delivered when the customer is trying to use a specific feature or is stuck on a certain task or prompt.

home card

An element displayed in the Alexa app to describe or enhance a voice interaction with a custom skill. See [Including a Card in Your Skill's Response](https://developer.amazon.com/en-US/docs/alexa/custom-skills/include-a-card-in-your-skills-response.html).

I

implicit confirmation (landmarking)

A prompt that subtly repeats back what Alexa heard to give the customer assurance that they were correctly understood. In the following example, repeating back the word horoscope is a landmarking technique used to establish trust with the customer but still supports natural dialog.  
  
For example, Alexa, ask Astrology Daily for my horoscope. Alexa would then ask to clarify the request with, Horoscope for what sign?

implicit grant

A grant type for requesting the end user's authorization to access their information in another system. In implicit grant, the authorization server returns the [access token](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#access-token) once the user logs in. This grant type is limited compared to [authorization code grant](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#auth-code-grant).

intents

A representation of the action that fulfills a customer's spoken request. Intents can have further arguments called [slots](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#slot) that represent variable information. For example, Alexa, ask History Buff what happened on June third. In this statement, …what happened on June third maps to a specific intent that can be handled by a particular Alexa ability. This tells Alexa that the user wants the skill History Buff to get historical information on a specific date.  
  
For details about defining intents, see [Create Intents, Utterances, and Slots](https://developer.amazon.com/en-US/docs/alexa/custom-skills/create-intents-utterances-and-slots.html).

intent schema

A JSON structure which declares the intents that can be handled by the service for a custom skill. This is documented in the [Interaction Model Schema](https://developer.amazon.com/en-US/docs/alexa/smapi/interaction-model-schema.html). You can also build your intent schema graphically in the [developer console](https://developer.amazon.com/en-US/docs/alexa/custom-skills/create-intents-utterances-and-slots.html).

interaction

An exchange of information in conversational format between the user and Alexa. This may be a single request-response, or a more extended set of turns.

interaction model

A defined set of logic parameters for your skill that will determine the voice interface through which customers will interact with your skill as well as map customers' spoken input to the intents your cloud-based service can handle.

interruptions

When the interaction between Alexa and the customer is interrupted by another event. Examples of interruptions are alarms and timers going off while the customer is talking to Alexa.

invocation

The act of beginning an interaction with a particular Alexa ability. For example, if a customer wants to wake Alexa to use the Horoscope skill, Alexa, ask Horoscope for today's reading.  
Alexa then follows up after the invocation and asks, What horoscope sign would you like?  
  
Types of invocations include: [full intent invocation](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#full-intent), [partial intent invocation](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#partial-intent), and [no intent invocation](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#no-intent).

invocation name

A name that represents the custom skill the customer wants to use. The customer says a supported phrase in combination with the invocation name for a skill to begin interacting with that skill. For example, Alexa, ask History Buff what happened on June 3rd. In this example, History Buff is the invocation name for a skill that retrieves historical events. Note that smart home skills do not have invocation names.

L

Lambda blueprint

An option in the AWS Lambda console that provides sample code and a sample configuration for a new Lambda function. There are multiple blueprints relevant to the Alexa Skills Kit. For example:alexa-smart-home-skill-adapter provides a sample skill for a smart home skill in Node.js. alexa-skills-kit-color-expert and alexa-skills-kit-color-expert-python provide samples of a simple custom skill in both Node.js and Python. See [Creating an AWS Lambda Function for a Custom Skill](https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-an-aws-lambda-function.html).

Lambda function

See [AWS Lambda function](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#aws-lambda-function).

landmark

See [Implicit confirmation](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#implicit-confirmation).

link account card

A special type of card (LinkAccount) displayed in the Alexa App that tells the user to link their account. The user can start the account linking process right from this card. A custom skill can [respond with this card](https://developer.amazon.com/en-US/docs/alexa/account-linking/add-account-linking-logic-custom-skill.html#respond-if-missing) if a user who has not linked their account tries to invoke an intent that requires authentication.

low confidence errors

When Alexa has low confidence that she correctly understood what the customer said. When this occurs, Alexa cannot proceed in the interaction without asking the question again or ending the interaction.

M

marker

Conversational markers, or cues, are words or sentence clauses that indicate the status of a dialog to a customer. For example, “First,” “Next,” “I'll need some information from you,” etc.

max error condition

When consecutive dialogue errors occur, this terminates the interaction and is designed to keep Alexa from making the same mistake repeatedly.

menu style prompt

A prompt that asks the customer a question intended to elicit a response from a small set of possible options (recommended 5 or fewer). For example, Helper for Minecraft. You can ask for a recipe, the ingredients of a potion, or game instructions. Now, which would you like?

N

natural language understanding (NLU)

A technology that helps computers to deduce what speakers actually mean, not just what words they say. NLU makes it possible for Alexa to infer, for example, that you're asking for a local weather forecast when you ask, Alexa, what's it like outside?

no intent invocation

A customer's request with no intent or slot information. For example, Alexa, open History Buff.

notifications

When the customer takes an action that requires Alexa to inform them at a later time that an event is occurring or about to occur. The most common examples of this are alarms and timers. A notification can occur if nothing else is going on at all, or they also can occur in the form of an interruption while the customer is interacting with Alexa. In this case, the notification is delivered between turns during an interaction.

O

one-time purchase

An [in-skill purchase](https://developer.amazon.com/en-US/docs/alexa/in-skill-purchase/isp-overview.html) (entitlement) that unlocks access to features or content within a skill. One-time purchases do not expire.

open ended prompt

A prompt that asks the customer a question intended to elicit a wide range of responses. For example, What would you like to do?

optional slot

A slot that contains values that refine the user's request, but are not necessary for Alexa to complete the task. For example, Alexa, ask History Buff what happened in history on June third. Here, …June third is optional since History Buff can just give historical events for today if the user does not specify a date. As such, you should not ask the user for optional slot values if they exclude them.

P

partial intent invocation

A customer's request that contains the customer's intent, but is missing a required slot. For example, Alexa, ask Horoscope for today's reading. Here, the required zodiac sign is missing, and Alexa needs to obtain that information from the customer.

prompt

A string of text that should be spoken to the customer to ask for more information. You include the prompt text in your response to a customer's request. Types of prompts include: [Open ended](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#open-ended), [Menu style](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#menu-style), [Re-prompt](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#re-prompt), and [Implicit confirmation (Landmarking)](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#implicit-confirmation).

R

redirect URL

An Amazon-provided URL to which the Authorization URI redirects the user after the user is authenticated. The redirect URLs are displayed in the developer portal after you configure account linking.

refresh token

A credential that can be used to obtain an access token. Refresh tokens are supported with the [authorization code grant type](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#auth-code-grant). A refresh token can be used to request a new access token after the old token expires.

re-prompt

A special kind of prompt used by Alexa when a response is not heard or clearly understandable, usually in the form of a question after a dialog error has occurred. The general purpose of a re-prompt is to help the customer recover from errors. For example, Alexa, open Score Keeper.  
Welcome back to Score Keeper. What's your update?  
  
In this case, if the customer doesn't respond, Alexa would continue with helpful tips. For example,  
You can add points for a player, ask for the current score, or start a new game. To hear a list of everything you can do, say Help. Now, what would you like to do?

required slot

A slot that contains values that are necessary for Alexa to complete the user's request. For example, Alexa, ask Astrology Daily for the horoscope for Taurus. Without the name of the specific zodiac sign, Astrology Daily cannot provide a horoscope. If the user does not provide a value for a required slot, you must ask the user for that slot value.

resource (APL)

In Alexa Presentation Language (APL), a named constant that you can use to specify a value instead of hardcoding values. For example, you can create a resource called myRed that defines a particular shade of red. In your APL document, you can then use @myRed for any property that accepts a color. A resource can have conditional logic, so it can resolve to different values based on criteria such as the viewport characteristics. To use the resource, prefix the resource name with the @ sign, as shown in the @myRed example.

resource owner

The user of your skill who can grant permission to access a protected resource (such as, but not limited to, account information) to connect your skill with their user account in another system (resource server). This is one of the four roles defined by OAuth 2.0, used in account linking.

resource server

The server hosting the resources the skill wants to access on behalf of the resource owner. One of the four roles defined by OAuth 2.0, used in account linking.  
For example, if your skill connects to a service for ordering rides (Car Hailer), the resource server might be a database containing profiles for Car Hailer users. If your skill is a smart home skill for controlling lighting (My Lights), the resource server is the part of your device cloud containing information about each user's smart devices that the service controls.

responsive component

An APL layout that combines components into modular, responsive elements. You can use these as building blocks in your document, similar to how you would use components. For example, [Button](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-alexa-button-layout.html) displays a button that works on all different types of devices and handles different button states automatically. A standard set of responsive components are available in the alexa-layouts package.

responsive template

An APL layout that provides a complete viewport design that combines components and responsive components. The template fills the entire viewport and includes the background, header, and content. For example, the [Text List](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-alexa-text-list-layout.html) template presents a scrolling list of text items with a background and header. A standard set of responsive templates are available in the alexa-layouts package.

S

sample utterance

A structured string of text that connects a specific intent to a likely [utterance](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#utterance). You provide a set of sample utterances as part of your [interaction model](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#interaction-model) for a custom skill. When customers say one of these utterances, the Alexa service sends a request to your service that includes the corresponding intent.

service

A cloud-based service you create to support a skill. This service takes requests from Alexa and returns responses. For a custom skill, the service accepts requests with [intents](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#intent) and returns responses with the text to speak back to the user. For a smart home skill, the service takes [device directives](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#device-directive), communicates with the device cloud to control devices such as lights and thermostats and sends [device events](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#device-event) back to Alexa.  
  
You can deploy the service for a custom skill either as an [AWS Lambda function](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#aws-lambda-function) or a [web service](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#web-service). Smart home skills can only be hosted using Lambda.

skill

A set of actions or tasks that are accomplished by Alexa. Skills are like apps for Alexa, helping customers perform everyday tasks or engage with your content naturally with voice. Alexa provides built-in functionality, such as timers and alarms. You can use the Alexa Skills Kit to create skills for Alexa. A skill includes both the code, in the form of a cloud-based service, and the configuration provided on the developer console.

slot

An argument to an intent that gives Alexa more information about that request. For example, Alexa, ask History Buff what happened on June third. In this statement, …June third is the value of a date slot that refines the request. For a given intent, slots can be [required](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#required-slot) or [optional](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#optional-slot). To designate slots as required and let Alexa manage the conversation to collect the slot values, [create a dialog model](https://developer.amazon.com/en-US/docs/alexa/custom-skills/define-the-dialog-to-collect-and-confirm-required-information.html) and then [delegate the dialog to Alexa](https://developer.amazon.com/en-US/docs/alexa/custom-skills/delegate-dialog-to-alexa.html).

Smart Home Skill API

An API to create skills that give Alexa the ability to control smart home devices such as lights and switches. The Smart Home Skill API translates utterances such as turn on the lights into [device directives](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#device-directive) that it routes to a Lambda function that can control a cloud-enabled device.

smart home endpoint

An endpoint identifies the target for a smart home directive and the origin of an event from a smart home skill. A smart home endpoint can represent a physical device, virtual device, group or cluster of devices, or a software component.

smart home skill

A skill intended to control smart home devices such as lights and thermostats. When using the Smart Home Skill API, the API defines the requests the skill can handle, such as [device directives](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#device-directive), and the words the users say to make those requests.  
  
A complete smart home skill includes the code hosted as an AWS Lambda function and a configuration that provides the information the Alexa service needs to route requests to the Lambda function. The code for the skill must be able to control the device (such as a light) using the cloud.

Speech Synthesis Markup Language (SSML)

A markup language that provides a standard way to mark up text to generate synthetic speech. The Alexa Skills Kit supports a subset of tags defined in the SSML specification.

subscription

An [in-skill purchase](https://developer.amazon.com/en-US/docs/alexa/in-skill-purchase/isp-overview.html) that offers access to premium content or features for a period of time. Users are charged on a recurring basis until they cancel their subscription.

system error

Occurs when something unexpected happened, unrelated to the dialogue between the customer and Alexa. For example, the call for a data service used to get the information the customer requested was unable to send Alexa that information.

T

tell

One of the words a user can say to tell Alexa to invoke a particular custom skill. This is used in combination with the [invocation name](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#invocation-name) for the skill. For example, Alexa, tell Color Expert that my favorite color is red. There are several phrases users can say to start a conversation with Alexa. See [Understanding How Users Invoke Custom Skills](https://developer.amazon.com/en-US/docs/alexa/custom-skills/understanding-how-users-invoke-custom-skills.html).

text-to-speech (TTS)

Converts a string of text to synthesized speech (Alexa's voice). The Alexa service can take plain text for TTS conversion.

timeouts/silence/no response errors

When the customer does not respond to a question Alexa asked. A re-prompt is usually played to encourage the customer to respond.

touch interaction

A touch on an Alexa-enabled device with a screen that produces a specified response, such as touching an item in a list on the screen to see more information about the item.

turn

A single request to or a response from Alexa. Sometimes turn refers to only the request side of a conversation, for example:  
  
Alexa, open Horoscope.  
What horoscope sign would you like?  
Pisces.  
Today's horoscope for Pisces is…  
  
This example might be referred to as a two-turn interaction, rather than the 4 turns that it technically contained.

U

utterance

The words the customer says to Alexa to convey what they want to do, or to provide a response to a question Alexa asks. For custom skills, you provide a set of sample utterances mapped to [intents](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#intent) as part of your custom [interaction model](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#interaction-model). For smart home skills, the [smart home skill API Message Reference](https://developer.amazon.com/en-US/docs/alexa/smarthome/smart-home-skill-api-message-reference.html) provides a predefined set of utterances.

V

viewport

On an Alexa-enabled device with a screen, the viewport is the area of the screen that the user can see. A viewport has a shape, orientation, size, and density.

viewport profile

A named resource that represents a range of viewport characteristics. The profile specifies a shape, orientation, range of sizes, and range of densities, so the profile can represent multiple physical devices. For example, the viewport profile hubLandscapeSmall encompasses rectangular, landscape devices where the width is between 960 and 1280 pixels, and the height is less than 600 pixels. Use the provided [Alexa viewport profiles](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-alexa-viewport-profiles-package.html) in conditional logic to [create responsive APL documents](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-build-responsive-apl-documents.html).

voice forward

A voice-optimized multimodal concept where the [GUI](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#gui) is optimized for voice. This means that buttons, links, and other touch affordances are replaced by voice-friendly affordances like action hints.

voice user interface (VUI)

A method for people to use voice input to interact with and control computers and devices. For a custom skill, the voice interface consists of a mapping between users' spoken utterances and the [intents](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#intent) your cloud-based service can handle. See [Interaction Model Schema](https://developer.amazon.com/en-US/docs/alexa/smapi/interaction-model-schema.html).

W

wake word

A command that the customer says to tell Alexa that they want to talk to her. For example, Alexa, open History Buff. Here, Alexa is the wake word. Alexa customers can select from a defined set of wake words: Alexa, Amazon, Echo, and Computer.

web service

In the context of the Alexa Skills Kit, an internet-accessible service that can accept requests from the Alexa service and return responses. You can use a web service as the [cloud-based service](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/alexa-skills-kit-glossary.html#cloud-based-service) for a custom skill. For the requirements your service must meet, see [Host a Custom Skill as a Web Service](https://developer.amazon.com/en-US/docs/alexa/custom-skills/host-a-custom-skill-as-a-web-service.html).

welcome message

The message delivered when a customer invokes a skill without a request. For example, Alexa, open Horoscope. Alexa would respond with a welcome message, such as: Welcome back to Horoscope. Which sign would you like to hear a daily horoscope for?  
  
A first time welcome message introduces the customer to the skill and provides a few commands the customer can use in that skill. A return welcome message greets the customer and prompts for a widely used task or required piece of information to continue the conversation.

# Latest Tips on Using the Documentation

Updated July 2019

Welcome to the Alexa Skills Kit (ASK) technical documentation. The technical documentation contains overviews, API reference, and focused code examples. This page provides the latest guidance to help you get the most out of the documentation.

* [New and updated documentation for Alexa Presentation Language (APL)](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#new-and-updated-documentation-for-alexa-presentation-language-apl)
* [New documentation for sound library](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#new-documentation-for-sound-library)
* [Updated documentation for request handling](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#updated-documentation-for-request-handling)
* [Code examples for you to copy](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#code-examples-for-you-to-copy)
* [Utterances in multiple languages](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#utterances-in-multiple-languages)
* [Quick reference pages to find relevant resources](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#quick-reference)
* [Feedback button](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#feedback)
* [Documentation for new features](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#documentation-for-new-features)
* [Other Alexa Skills Kit resources](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#other-alexa-skills-kit-resources)

## New and updated documentation for Alexa Presentation Language (APL)

Use [Alexa Presentation Language (APL)](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/understand-apl.html) to create visual experiences for your Alexa skill. See the following new and updated documentation for the latest information about APL.

* [Build Responsive APL Documents](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-build-responsive-apl-documents.html) – New documentation about best practices for building responsive APL documents, so your skill's visuals work on different sized devices.
* [APL Document (reference)](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-document.html) – Reorganized reference documentation to make information easier to find.
* [Changes Introduced in APL Version 1.1](https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-latest-version.html) – Read about the new features and capabilities introduced in APL 1.1, the latest version of APL.

## New documentation for sound library

The documentation page for the [Alexa Skills Kit Sound Library](https://developer.amazon.com/en-US/docs/alexa/custom-skills/ask-soundlibrary.html) now lists all of the available sounds on a single page. The sound library now contains more than 3,000 sounds, and you can use the new documentation page to search, filter, and listen to them. You can also copy the [Speech Synthesis Markup Language (SSML)](https://developer.amazon.com/en-US/docs/alexa/custom-skills/speech-synthesis-markup-language-ssml-reference.html) code to use a sound in your skill.

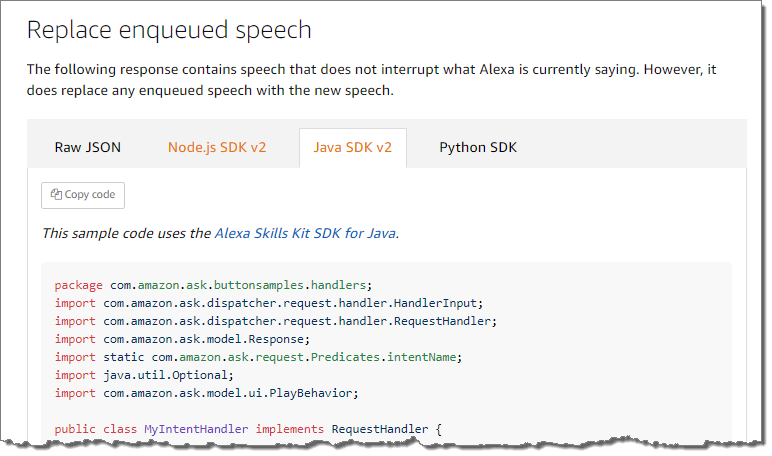
## Updated documentation for request handling

The topics that cover basic request handling and session management have been substantially revised. The docs now include code samples for use with the Node, Java, and Python SDKs. See the updated docs here:

* [Handle Requests Sent by Alexa](https://developer.amazon.com/en-US/docs/alexa/custom-skills/handle-requests-sent-by-alexa.html)
* [Manage the Skill Session and Session Attributes](https://developer.amazon.com/en-US/docs/alexa/custom-skills/manage-skill-session-and-session-attributes.html)

## Code examples for you to copy

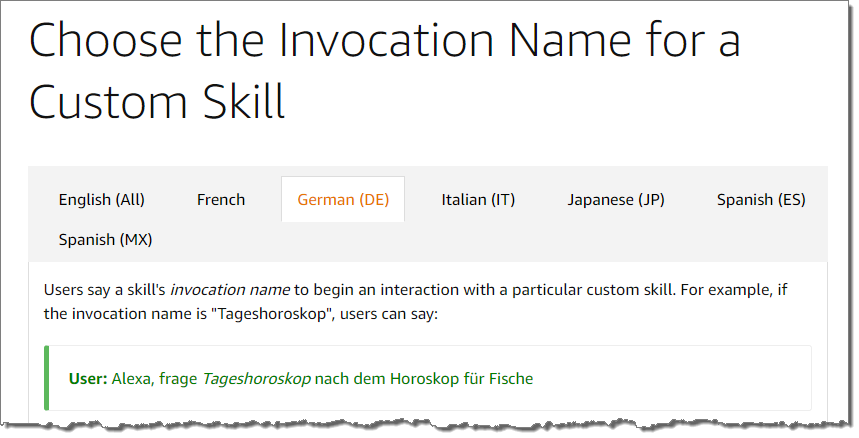
In the documentation, look for short, focused code examples that demonstrate features using the [Alexa Skills Kit SDKs](https://developer.amazon.com/en-US/docs/alexa/quick-reference/use-sdks-quick-reference.html). The code examples focus on specific features that you can incorporate into your own code with minimal modification. The following is a code example from [Control Interruption of Alexa's Speech](https://developer.amazon.com/en-US/docs/alexa/echo-button-skills/control-interruption-alexa-speech.html#speech-replace-enqueued-example).

[](https://images-na.ssl-images-amazon.com/images/G/01/mobile-apps/dex/ask-tech-docs/code-example._TTH_.png)

If you can't find a code example for the feature that you want to add to your skill, please let us know using the [feedback button](https://developer.amazon.com/en-US/docs/alexa/ask-overviews/latest-tips-documentation.html#feedback).

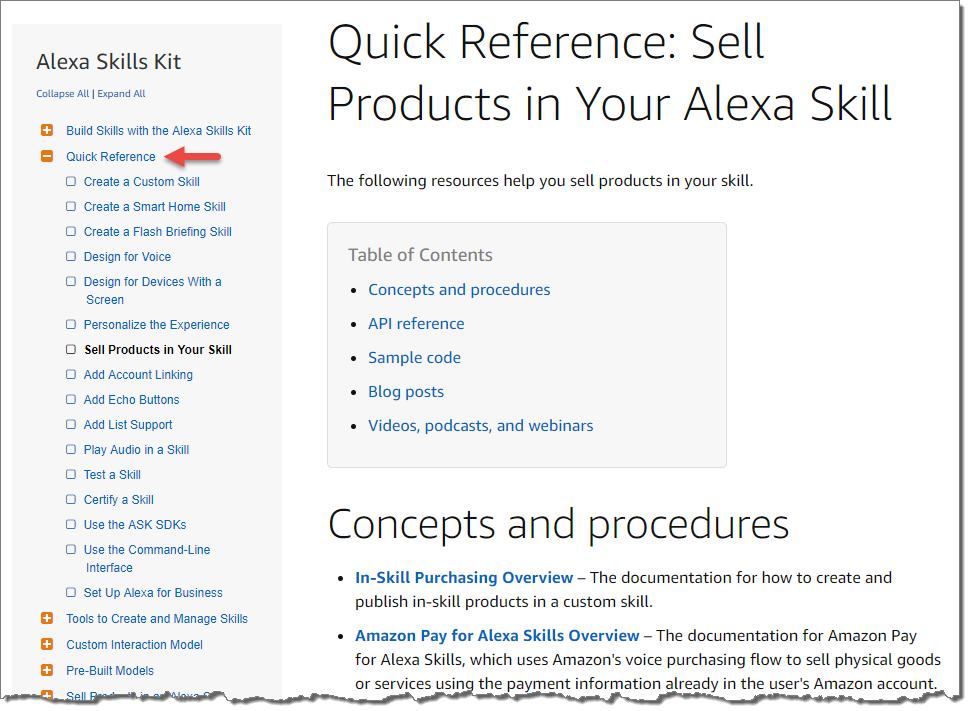
## Utterances in multiple languages

When the documentation provides an example of a user utterance, you can often find a language tab for each supported language. The following is an example from [Choose the Invocation Name for a Custom Skill](https://developer.amazon.com/en-US/docs/alexa/custom-skills/choose-the-invocation-name-for-a-custom-skill.html).

[](https://images-na.ssl-images-amazon.com/images/G/01/mobile-apps/dex/ask-tech-docs/choose-invocation-name-skill._TTH_.png)

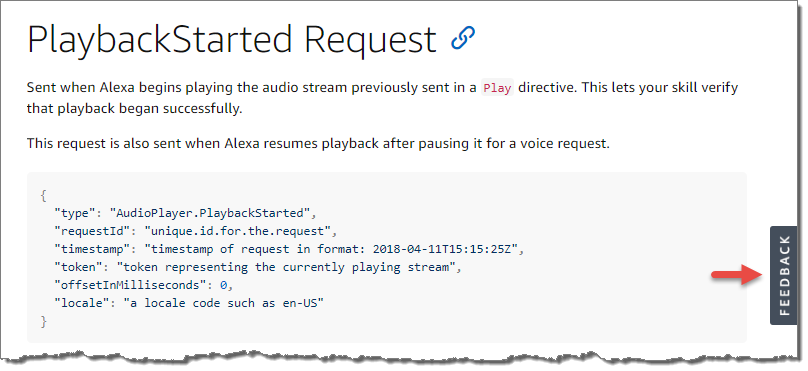
## Quick reference pages to find relevant resources

The quick reference pages provide consolidated lists of various resources – concepts, procedures, API reference, sample code, blog posts, videos, podcasts, webinars, Twitch streams – related to a specific topic, such as [selling products in your skill](https://developer.amazon.com/en-US/docs/alexa/quick-reference/sell-products-quick-reference.html).

[](https://images-na.ssl-images-amazon.com/images/G/01/mobile-apps/dex/ask-tech-docs/qr-page-sell-products._TTH_.png)

## Feedback button

You can help us improve our documentation! Provide feedback by using the feedback button that appears on every page.

[](https://images-na.ssl-images-amazon.com/images/G/01/mobile-apps/dex/ask-tech-docs/feedback-button._TTH_.png)

## Documentation for new features

You can find links to the technical documentation for the latest Alexa Skills Kit features at [Release Update](https://developer.amazon.com/alexa/feature-updates).