- VideoApp Interface Reference
- Speech Synthesis Markup Language (SSML) Reference
- Sell Products in an Alexa Skill
- Smart Home Skills
- Works with Alexa Certification
- Alexa for Business
- Capability Interface Reference
- Video Skills
- Flash Briefing Skills
- List Skills
- Gadget Skills
- Account Linking
- ASK SDKs
- ASK CLI and Skill Management API
- Frequently Asked Questions

Other Resources

- Alexa Skills Kit Forum
- Alexa Voice Service
- Alexa Fund

Dialog Interface Reference

The `Dialog` interface provides directives for managing a *multi-turn conversation* between your skill and the user. You can use the directives to ask the user for the information you need to fulfill their request.

Table of Contents

Dialog Directive Requirements

To use the `Dialog` directives, your skill must meet these requirements:

- You must include a dialog model. This identifies required slots and the prompts and utterances for these slots.
- The interaction model cannot include the `AMAZON.LITERAL` built-in slot type.

Dialog Model

To use any of the `Dialog` directives, you must create a dialog model This structure identifies:

- The slots that *must* be filled with valid values in order to fulfill the intent. These are considered *required slots*.
- The prompts Alexa speaks to ask for required slot values and the utterances users can say in reply.
- Whether any of the required slots must be also confirmed by the user before continuing.
- Whether the entire intent must be confirmed by the user before continuing.
- The prompts Alexa speaks to ask for slot and intent confirmations.

How the dialog model is used when a user interacts with your skill depends on how you choose tomanage the conversation.

Steps of a Multi-turn Dialog or Conversation

In an Alexa skill, a *dialog* with the user is a *conversation with multiple turns* in which Alexa asks questions and the user responds with the answers. The conversation is tied to a specific *intent* representing the user's overall request. The questions and answers are intended to gather and confirm values for all of the intent's *required slots*. The conversation continues until all slots needed for the intent are filled and confirmed.

The questions Alexa asks during the conversation fall into three categories:

- *Slot elicitation*: Ask the user for a slot value. The user answers with a slot value or a full utterance including the slot value. Examples of slot elicitation questions:
  - **Alexa:** What city are you leaving from? *(Eliciting the value for a `fromCity` slot.)*
  - **Alexa:** Where are you traveling to? *(Eliciting the value for a `toCity` slot.)*
  - **Alexa:** When did you want to travel? *(Eliciting the value for a `travelDate` slot.)*
- *Slot confirmation*: Ask the user to confirm that a *single slot value* previously provided (or set programmatically) is correct. The user answers with "yes" or "no". Examples of slot confirmation questions:
  - **Alexa:** You said you're leaving from *Seattle*, right? *(Confirming the `fromCity` value.)*
  - **Alexa:** Did you want to travel to *Portland*? *(Confirming the `toCity` value.)*
  - **Alexa:** You're traveling on *April 21st*, right? *(Confirming the `travelDate` value.)*
- *Intent confirmation*: Ask the user to verify that *all* the information gathered for the intent is correct before fulfilling the intent. As with slot confirmation, the user answers with "yes" or "no". Examples of intent confirmation questions:
  - **Alexa:** I'm saving your trip from *Seattle* to *Portland* on *April 21st*. Is that OK? *(Confirming the entire `PlanMyTrip` intent.)*

The dialog for a particular intent might include all of these steps or only some of them. For instance, a dialog could include slot elicitation, but not use slot or intent confirmation.

About Managing the Conversation with the User

There are three main scenarios for handling a multi-turn conversation with the user in your skill:

1. Let Alexa use the prompts defined in the *dialog model* by returning the `Dialog.Delegate` directive.
2. Control each step of the dialog yourself using `Dialog.ElicitSlot`, `Dialog.ConfirmSlot`, and `Dialog.ConfirmIntent`.
3. Combine both options – delegate on some turns, but take control on others when necessary.

Delegate the Dialog to Alexa

This option lets you focus most of your coding efforts on the logic for fulfilling the user's request, rather than writing all the code to ask the user for slot values yourself.

When Alexa sends your skill an intent, the `IntentRequest` includes a `dialogState` property that indicates whether there may be additional steps in the dialog (`STARTED` or `IN_PROGRESS`) or if all the steps are complete and the `IntentRequest` has all the required slot values and confirmations from the user (`COMPLETED`).

If the dialog is not yet complete, you return the `Dialog.Delegate` directive. Alexa determines the next step in the dialog and uses the prompts defined in the dialog model to elicit slot values, confirm slot values, or confirm the entire intent.

Note: Your skill gets an `IntentRequest` for *every turn* of the conversation. At any point, you can take over the dialog rather than continuing to delegate to Alexa. This can be useful if you need to default or calculate slot values based on the information you have so far, rather than stepping through the entire dialog.

Once the conversation is complete, the incoming `IntentRequest` has a `dialogState` of `COMPLETED`. All required information is now available in the intent's slot values. Your skill can fulfill the user's request at this point.

Control the Dialog in Your Skill Code

In this option, your code checks for slot values and confirmation status, determines the next step in the conversation, and returns the appropriate directives (`Dialog.ElicitSlot`, `Dialog.ConfirmSlot`, or `Dialog.ConfirmIntent`).

You need to determine the status and next steps in your own code. If your skill meets the requirements to use the `Dialog` directives, the `IntentRequest` sent to your skill does include `dialogState`. However, this is set to either `STARTED` (when the intent is invoked) or `IN_PROGRESS`. The `COMPLETED` status is only possible when you use `Dialog.Delegate`.

Note that the directives do not use the prompts defined in your dialog model in this scenario. For instance, when you return `Dialog.ElicitSlot`, you must include the prompt in your response. `Dialog.ElicitSlot` *does* use the utterances you provide for the slot. Alexa biases the interaction model to listen for the utterances defined for the slot, so it is important to provide good utterances when you define the dialog model.

Both Delegate and Control the Dialog Manually

You can combine the two main options. Depending on the incoming `IntentRequest`, your code would take one of these actions:

- Return `Dialog.Delegate` to let Alexa handle the next step.
- Return one of the other directives (such as `Dialog.ElicitSlot`) to take control of the dialog yourself.

Update Slot Values and Confirmation Status During the Dialog

Each of the `Dialog` directives includes an `updatedIntent` property that can take an `Intent` object. Use this to:

- Set or change any slot values in code before continuing with the dialog. This can reduce the number of questions Alexa must ask to fulfill the intent, which improves the user experience.

  For instance, you can set slot values based on user defaults or other information you are persisting. Combine this with the `Dialog.Delegate` or `Dialog.ConfirmSlot` directives.

- Set or change the `confirmationStatus` for a slot or for the entire intent.

You cannot change intents when returning a `Dialog` directive, so the intent name and full set of slots must match the intent sent to your skill.

Dialog Directives

The following table summarizes the `Dialog` directives. See the sections below for details about each directive.

If your skill does not meet the requirements to use the `Dialog` directives, returning any of these directives causes an error.

Directive	Description
`Dialog.Delegate`	Sends Alexa a command to handle the next turn in the dialog with the user. You can use this directive if the skill has a *dialog model*

Directive	Description
	and the current status of the dialog (dialogState) is either STARTED or IN_PROGRESS . You cannot return this directive if the dialogState is COMPLETED .
Dialog.ElicitSlot	Sends Alexa a command to ask the user for the value of a specific slot. Specify the name of the slot to elicit in the slotToElicit property. Provide a prompt to ask the user for the slot value in an OutputSpeech object in the response.
Dialog.ConfirmSlot	Sends Alexa a command to confirm the value of a specific slot before continuing with the dialog. Specify the name of the slot to confirm in the slotToConfirm property. Provide a prompt to ask the user for confirmation in an OutputSpeech object in the response. Be sure repeat back the value to confirm in the prompt.
Dialog.ConfirmIntent	Sends Alexa a command to confirm the *all the information* the user has provided for the intent before the skill takes action. Provide a prompt to ask the user for confirmation in an OutputSpeech object in the response. Be sure to repeat back all the values the user needs to confirm in the prompt.

Delegate Directive

Sends Alexa a command to handle the next turn in the dialog with the user. You can use this directive if the skill has a *dialog model* and the current status of the dialog (dialogState) is either STARTED or IN_PROGRESS. You cannot return this directive if the dialogState is COMPLETED.

Syntax

```
{
  "type": "Dialog.Delegate",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

Parameter	Description	Type	Required
type	Set to Dialog.Delegate .	string	Yes
updatedIntent	An intent object representing the intent sent to your skill. You can use this property set or change slot values and confirmation status if necessary.	object	No

Parameter	Description	Type	Required
	If you don't need to change any slot values or confirmation statuses, you can leave this property out of your response. Note that you cannot change intents when returning a Dialog directive, so the intent name and set of slots must match the intent sent to your skill.		

Details

Alexa determines the next step in the dialog based on the dialog model. In a dialog for an intent with multiple slots, Alexa elicits and confirms (if necessary) each slot in the order defined in the dialog model. Intent confirmation (if necessary) happens once all required slots have values.

Do not include outputSpeech or reprompt with the Dialog.Directive. Alexa uses the prompts defined in the dialog model to ask the user for the slot values and confirmations.

Use the updatedIntent property to set or change the slot values and/or the confirmation status. You can leave this property out to continue the dialog using the same slot values and statuses that triggered the incoming IntentRequest.

For example, you might want to pre-fill certain slots with default values on the first turn, then return Delegate to let Alexa proceed with the dialog:

1. Check the dialog status with the dialogState property on the IntentRequest.
2. If the dialog is just starting (dialogState is STARTED), update the slots on the Intent object with default values as needed. Include this updated intent in the updatedIntent property on the Dialog.Delegate directive.
3. If the dialog is IN_PROGRESS, return Dialog.Delegate with no updatedIntent.
4. Once the dialog is COMPLETED, do your normal intent handling.

```
// The intentRequest variable here is the IntentRequest object sent to the skill.
if (intentRequest.dialogState === "STARTED"){
    // Pre-fill slots: update the intent object with slot values for which
    // you have defaults, then return Dialog.Delegate with this updated intent
    // in the updatedIntent property.
} else if (intentRequest.dialogState != "COMPLETED"){
    // return a Dialog.Delegate directive with no updatedIntent property.
} else {
    // Dialog is now complete and all required slots should be filled,
    // so call your normal intent handler.
    handlePlanMyTripIntent(intent, session, callback);
}
```

⚠ **Important:** If your dialog model for the intent includes prompts for intent confirmation, your code should *also* make sure that the Intent.confirmationStatus property is CONFIRMED before fulfilling the user's request. If the user responds to the confirmation prompt with 'no', the dialogState is COMPLETED and the confirmationStatus is DENIED.

Example Interaction

An intent for planning a trip (PlanMyTrip) could have an interaction like the following:

User: Alexa, tell *Plan My Trip* I want to visit Portland.
Alexa sends the skill a `PlanMyTrip` intent with:

- `dialogState` : `STARTED`
- `fromCity` : `null`
- `toCity` : `Portland`
- `travelDate` : `null`
- `travelMode` : `null`
- `activity` : `null`
- `confirmationStatus` : `NONE`

The skill returns a `Dialog.Delegate` directive with: `updatedIntent.slots.fromCity` : `Seattle` and the other slots unchanged (this defaults the `fromCity` to a value)

Plan My Trip: You wanted to fly out of Seattle, right? *(Confirmation prompt for the `fromCity` slot, as defined in the dialog model.)*
User: Yes.
Alexa sends the `PlanMyTrip` intent with:

- `dialogState` : `IN_PROGRESS`
- `fromCity` : `Seattle` (now with `confirmationStatus` : `CONFIRMED`)
- `toCity` : `Portland`
- `travelDate` : `null`
- `travelMode` : `null`
- `activity` : `null`
- `confirmationStatus` : `NONE`

Skill returns a `Dialog.Delegate` directive.

Alexa: When did you want to travel? *(This elicitation prompt for `travelDate` comes from the dialog model.)*
User: On Friday
Alexa sends the `PlanMyTrip` intent with:

- `dialogState` : `IN_PROGRESS`
- `fromCity` : `Seattle` (`confirmationStatus` : `CONFIRMED`)
- `toCity` : `Portland`
- `travelDate` : `2017-04-21`
- `travelMode` : `null`
- `activity` : `null`
- `confirmationStatus` : `NONE`

Skill returns a `Dialog.Delegate` directive. (in this example, the `travelMode` and `activity` slots are not required, so Alexa does not prompt for them.)

Alexa: I'm saving your trip from Seattle to Portland on April 21st. Is that OK? *(Intent confirmation prompt from the dialog model.)*
User: Yes.
Alexa sends the `PlanMyTrip` intent with:

- `dialogState` : `COMPLETED`
- `fromCity` : `Seattle` (`confirmationStatus` : `CONFIRMED`)
- `toCity` : `Portland`
- `travelDate` : `2017-04-21`
- `travelMode` : `null`
- `activity` : `null`
- `confirmationStatus` : `CONFIRMED`

> *Skill now has all needed information and can handle the* PlanMyTrip *intent. Since the* dialogState *is* COMPLETED*, the skill can no longer return the* Dialog.Delegate *directive.*

If the user denied the confirmation in the final turn, the dialog would still be considered COMPLETE, but the intent.confirmationStatus would be DENIED:

> *...earlier interaction to invoke* PlanMyTrip *and set the* fromCity*,* toCity*, and* travelDate *slots.*
>
> **Alexa:** I'm saving your trip from Seattle to Portland on April 21st. Is that OK? *(Intent confirmation prompt from the dialog model.)*
> **User:** No.
> *Alexa sends the* PlanMyTrip *intent with:*
>
> - dialogState: COMPLETED
> - fromCity: Seattle (confirmationStatus: CONFIRMED)
> - toCity: Portland
> - travelDate: 2017-04-21
> - travelMode: null
> - activity: null
> - confirmationStatus: DENIED
>
> *The dialog is considered complete since all information was gathered. Since the* dialogState *is* COMPLETED*, the skill can no longer return the* Dialog.Delegate *directive.*

(Back up to Dialog Directives)

ElicitSlot Directive

Sends Alexa a command to ask the user for the value of a specific slot. Specify the name of the slot to elicit in the slotToElicit property. Provide a prompt to ask the user for the slot value in an OutputSpeech object in the response.

If your skill does not meet the requirements to use the Dialog directives, returning Dialog.ElicitSlot causes an error.

Syntax

Directive Syntax   Example

```
{
  "type": "Dialog.ElicitSlot",
  "slotToElicit": "string",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

Parameter	Description	Type	Required
`type`	Set to `Dialog.ElicitSlot`.	`string`	Yes
`slotToElicit`	The name of the slot to elicit.	`string`	Yes
`updatedIntent`	An intent object representing the intent sent to your skill. You can use this property set or change slot values and confirmation status if necessary. If you don't need to change any slot values or confirmation statuses, you can leave this property out of your response. Note that you cannot change intents when returning a `Dialog` directive, so the intent name and set of slots must match the intent sent to your skill.	`object`	No

Details

You must include the prompt to ask the user for the slot value in the `OutputSpeech` object. The directive does *not* use the prompts defined in the dialog model.

`Dialog.ElicitSlot` *does* use the utterances you provide for the slot. Alexa biases the interaction model to listen for the utterances defined for the slot, so it is important to provide good utterances when you define the dialog model.

If you need to set or change any slot values or confirmation statuses before continuing the dialog, set those values in an Intent object and pass that object in the `updatedIntent` property.

Example Interaction

An intent for planning a trip (`PlanMyTrip`) could have an interaction like the following:

> **User:** Alexa, tell *Plan My Trip* I want to go on a trip.
> *Alexa sends the skill a `PlanMyTrip` intent with no slot values:*
>
> - `fromCity` : `null`
> - `toCity` : `null`
> - `travelDate` : `null`
> - `travelMode` : `null`
> - `activity` : `null`
>
> *Skill sends a `Dialog.ElicitSlot` directive with `slotToElicit` set to `fromCity` .*
>
> **Plan My Trip:** From where did you want to start your trip? *(This elicitation prompt comes from the `outputSpeech` included in the response.)*
> **User:** Seattle.
> *Alexa sends the `PlanMyTrip` intent with:*
>
> - `fromCity` : `Seattle`
> - `toCity` : `null`
> - `travelDate` : `null`
> - `travelMode` : `null`
> - `activity` : `null`
>
> *Skill sends a `Dialog.ElicitSlot` directive with `slotToElicit` set to `toCity` .*
>
> **Plan My Trip:** Where are you traveling to?
> **User:** Portland.
> *Alexa sends the `PlanMyTrip` intent with:*
>
> - `fromCity` : `Seattle`
> - `toCity` : `Portland`
> - `travelDate` : `null`
> - `travelMode` : `null`
> - `activity` : `null`
>
> *(Dialog continues as the skill sends `ElicitSlot` directives to collect remaining slots for the intent.)*

(Back up to `Dialog` Directives)

ConfirmSlot Directive

Sends Alexa a command to confirm the value of a specific slot before continuing with the dialog. Specify the name of the slot to confirm in the `slotToConfirm` property. Provide a prompt to ask the user for confirmation in an `OutputSpeech` object in the response. Be sure repeat back the value to confirm in the prompt.

If your skill does not meet the requirements to use the `Dialog` directives, returning `Dialog.ConfirmSlot` causes an error.

Syntax

Directive Syntax | Example

```
{
  "type": "Dialog.ConfirmSlot",
  "slotToConfirm": "string",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

Parameter	Description	Type	Required
type	Set to `Dialog.ConfirmSlot` .	string	Yes
slotToConfirm	The name of the slot to confirm.	string	Yes
updatedIntent	An intent object representing the intent sent to your skill. You can use this property set or change slot values and confirmation status if necessary. If you don't need to change any slot values or confirmation statuses, you can leave this property out of your response. Note that you cannot change intents when returning a `Dialog` directive, so the intent name and set of slots must match the intent sent to your skill.	string	No

Details

☑ **Tip:** Use slot confirmation sparingly – avoid annoying the user with too many "are you sure" prompts during the interaction.

When returning `Dialog.ConfirmSlot` , you must include the prompt to ask the user for confirmation in the `OutputSpeech` object. The directive does *not* use the prompts defined in the dialog model.

If you need to change any slot values before asking the user to confirm the intent, set those values in an Intent object and pass that object in the `updatedIntent` property.

After Alexa asks the user to confirm the slot value, the user can respond with "yes" or "no." Alexa then sends your skill the intent with an updated `confirmationStatus` property on the specific slot to indicate the user's response (`CONFIRMED` or `DENIED`).

The user can deny the confirmation twice, after which the session ends.

Example Interactions

An intent for planning a trip (`PlanMyTrip`) could have an interaction like the following:

> **User:** Alexa, tell Plan My Trip to plan a trip from *Seattle*
> *Alexa sends the skill a `PlanMyTrip` intent with:*
>
> - `fromCity` : `Seattle` (`confirmationStatus` is `NONE` since the user has not confirmed this value)
> - `toCity` : `null`
> - `travelDate` : `null`
> - `travelMode` : `null`
> - `activity` : `null`
>
> *Skill sends a `Dialog.ConfirmSlot` directive with `slotToConfirm` set to `fromCity` .*
>
> **Plan My Trip:** You said you're leaving from Seattle, right? *This confirmation prompt comes from the `outputSpeech` included in the response.*
>
> **User:** Yes. *(The user's "Yes" response does not send a normal `AMAZON.YesIntent` , but instead sends the `PlanMyTrip` intent with the `fromCity` slot set to "Seattle" and `fromCity.confirmationStatus` set to `CONFIRMED` .)*

(Back up to `Dialog` Directives)

ConfirmIntent Directive

Sends Alexa a command to confirm the *all the information* the user has provided for the intent before the skill takes action. Provide a prompt to ask the user for confirmation in an `OutputSpeech` object in the response. Be sure to repeat back all the values the user needs to confirm in the prompt.

If your skill does not meet the requirements to use the `Dialog` directives, returning `Dialog.ConfirmIntent` causes an error.

Syntax

Directive Syntax   Example

```
{
  "type": "Dialog.ConfirmIntent",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

Parameter	Description	Type	Required
`type`	Set to `Dialog.ConfirmIntent`.	`string`	yes
`updatedIntent`	An intent object representing the intent sent to your skill. You can use this property set or change slot values and confirmation status if necessary. If you don't need to change any slot values or confirmation statuses, you can leave this property out of your response. Note that you cannot change intents when returning a `Dialog` directive, so the intent name and set of slots must match the intent sent to your skill.	`object`	No

Details

Use this directive after your skill has gathered all the required slots you need to fulfill an intent, but you want to give the user one more opportunity to confirm that it is correct before continuing. This is common for skills that order products or make reservations.

✓ **Tip:** Only use intent confirmation when necessary – avoid annoying the user with too many "are you sure" prompts during the interaction.

When returning `Dialog.ConfirmIntent`, you must include the prompt to ask the user for confirmation in the `OutputSpeech` object. The directive does *not* use the prompts defined in the dialog model.

If you need to change any slot values before asking the user to confirm the intent, set those values in an Intent object and pass that object in the `updatedIntent` property.

After Alexa asks the user to confirm the information, the user can respond with "yes" or "no." Alexa then sends your skill the intent with an updated `intent.confirmationStatus` property to indicate the user's response (`CONFIRMED` or `DENIED`).

Example Interaction

An intent for planning a trip (`PlanMyTrip`) could have an interaction like the following:

> *...Previous interactions already collected the required `fromCity` , `toCity` , and `travelDate` slots. Alexa sends the skill a `PlanMyTrip` intent with all the slots filled in, but the overall intent `confirmationStatus` set to `NONE` .*
>
> *Skill sends a `Dialog.ConfirmIntent` .*
>
> **Plan My Trip:** I'm saving your trip from Seattle to Portland on April 21st. Is that OK? *(This confirmation prompt comes from the `outputSpeech` included in the response.)*
> **User:** Yes. *(The user's 'Yes' response does not send a normal `AMAZON.YesIntent` , but instead sends the `PlanMyTrip` intent with all slots filled in and a `confirmationStatus` set to `CONFIRMED` .)*

(Back up to `Dialog` Directives)

Service Interface Reference (JSON)

Request Format and Standard Request Types:

- Request and Response JSON Reference
- Request Types Reference (LaunchRequest, IntentRequest, SessionEndedRequest)

Interfaces:

- AudioPlayer Interface Reference
- **Dialog Interface Reference** (this document)
- Display Interface Reference
- GadgetController Interface Reference
- GadgetController Interface Reference
- GameEngine Interface Reference
- GameEngine Interface Reference
- PlaybackController Interface Reference
- VideoApp Interface Reference