Alexa

Alexa Skills Kit

Alexa Voice Service

Alexa Fund

Smart Home Skill API Reference

Last updated April 7, 2017

Current Developers: See what's changed

Smart Home Skill API

- Smart Home Skill API Documentation
  - Understanding the Smart Home Skill API
  - Steps to Create a Smart Home Skill
  - **Smart Home Skill API Reference**
  - Smart Home Skill Publishing Guide
  - Developing Smart Home Skills in Multiple Languages
  - Providing Scenes in a Smart Home Skill
  - Linking an Alexa User with a User in Your System

JSON message format. The skill adapter is hosted as a Lambda function on AWS Lambda (a service offering by Amazon Web Services) with Alexa Connected Home configured as the event source. The service sends events as JSON-formatted requests that contain a header and payload object to your skill adapter. Your skill adapter returns a JSON-formatted response that also contains a header and payload. The syntax used for these requests and responses is the skill adapter *directive language*. An individual request or response is a skill adapter *directive*.

In the code for your skill adapter, you need to read the directive headers to determine the message type sent by the Smart Home Skill API. You will use details of the message payload to communicate with the customer's cloud-connected device, and respond indicating the action was successful or unsuccessful. Following are some details about directive structure.

Authentication

The Smart Home Skill API follows the OAuth2.0 specification. Every request sent from the Smart Home Skill API to a skill adapter contains an OAuth access token in the request. In addition, the device cloud for any cloud-enabled device must support the authorization code grant flow type.

Make sure that the third-party provider implementing OAuth has white-listed the OAuth redirect endpoint assigned to your smart home skill. You can find this URL listed as the **Redirect URL** on the **Configuration** page for your app on the developer portal. Also, the OAuth provider must have a certificate signed by an Amazon-approved certificate authority.

To learn more about how authentication in Alexa works, see Linking an Alexa User with a User in Your System.

Skill Adapter Directives

All skill adapter directives, whether sent by the Smart Home Skill API to your adapter or sent by your adapter back to the Smart Home Skill API, share the

- Smart Home Skill API Forum
- Skills Beta Testing
- Getting Started with the Alexa Skills Kit
- Custom Skills
- Flash Briefing Skills

- **Header**
- **Payload**

In addition, the size limit on the skill adapter directive in either direction is 128KB.

Message Headers

The directive header has a set of expected fields that are the same across message types. These describe the message namespace, the directive name, targeted version, and a unique message identifier. Following is example JSON for a typical message header:

```
{
    "header": {
        "messageId": "6d6d6e14-8aee-473e-8c24-0d31ff9c17a2",
        "name": "DiscoverAppliancesRequest",
        "namespace": "Alexa.ConnectedHome.Discovery",
        "payloadVersion": "2"
    }
}
```

A header must contain the following properties:

Property	Description
`messageID`	A unique identifier for a single request or response. This is used for tracking purposes and a skill adapter should log this information, although it should not be used to support business logic. Every message from a skill adapter must have this field populated. Any string of alphanumeric characters and dashes less than 128 characters is valid, but a version 4 UUID, which is a UUID generated from random numbers, is recommended.
`name`	The name of the directive such as `DiscoverAppliancesRequest` or `DiscoverAppliancesResponse`

payload. The current categories are:

- `Alexa.ConnectedHome.Discovery`
- `Alexa.ConnectedHome.Control`
- `Alexa.ConnectedHome.Query`

`payloadVersion`	The API version that should be applied to the payload message. The current version is 2, and this payload format is described in this document. Version 1 can be used to describe the previously released Alexa Lighting API.

Message Payload

The payload for a skill adapter directive depends on the name of the directive specified in the header and payload properties will vary depending on the directive contained in the request. Payload contents are described in detail with each directive type.

The following sections describe the different types of directives, their expected payload descriptions, and examples.

Task	Namespace	Message Names
Discover connected devices	Alexa.ConnectedHome.Discovery	▪ DiscoverAppliancesRequest ▪ DiscoverAppliancesResponse

	control connected devices; turn things off and on and change settings	Alexa.ConnectedHome.Control	▪ DecrementColorTemperatureRequest ▪ DecrementColorTemperatureConfirmation ▪ DecrementPercentageRequest ▪ DecrementPercentageConfirmation ▪ IncrementColorTemperatureRequest ▪ IncrementColorTemperatureConfirmation ▪ IncrementPercentageRequest ▪ IncrementPercentageConfirmation ▪ IncrementTargetTemperatureRequest ▪ IncrementTargetTemperatureConfirmation ▪ SetColorRequest ▪ SetColorConfirmation ▪ SetColorTemperatureRequest ▪ SetColorTemperatureConfirmation ▪ SetLockStateRequest ▪ SetLockStateConfirmation ▪ SetPercentageRequest ▪ SetPercentageConfirmation ▪ SetTargetTemperatureRequest ▪ SetTargetTemperatureConfirmation ▪ TurnOnRequest ▪ TurnOnConfirmation ▪ TurnOffRequest

Query connected devices for their current state	Alexa.ConnectedHome.Query	▪ GetLockStateRequest ▪ GetLockStateResponse ▪ GetTargetTemperatureRequest ▪ GetTargetTemperatureResponse ▪ GetTemperatureReadingRequest ▪ GetTemperatureReadingResponse

Discovery Messages

These message types identify the device and capabilities available to this skill adapter.

- DiscoverAppliancesRequest
- DiscoverAppliancesResponse

DiscoverAppliancesRequest

Example Utterances:

“Alexa, discover my smart home devices”

“Alexa, finde meine smarten geräte”

Purpose: Discover devices or scenes associated with the end-user’s device cloud account. A `DiscoverAppliancesRequest` is sent from Smart Home Skill API to the skill adapter with the goal of discovering devices or scenes associated with the customer’s device cloud account. For information on discovering scenes, see Providing Scenes in a Smart Home Skill. If there are no devices to discover or if your device cloud

Header

Property	Value
name	DiscoverAppliancesRequest
namespace	Alexa.ConnectedHome.Discovery

Payload

Property	Description	Required
accessToken	Access token associated with the customer's device cloud account	Yes

Examples

DiscoverAppliancesRequest example:

```
{
    "header": {
        "messageId": "6d6d6e14-8aee-473e-8c24-0d31ff9c17a2",
        "name": "DiscoverAppliancesRequest",
        "namespace": "Alexa.ConnectedHome.Discovery",
        "payloadVersion": "2"
    },
    "payload": {
        "accessToken": "*OAuth Token here*"
    }
}
```

Return To Top

Purpose: Returns all of the devices associated with the end-user's device cloud account. The expected response from the skill adapter to the Smart Home Skill API for a `DiscoverAppliancesRequest`. If there are no devices to discover or if your device cloud encounters an error, return an empty `discoveredAppliances` array.

Header

Property	Value
name	DiscoverAppliancesResponse
namespace	Alexa.ConnectedHome.Discovery

Payload

Property	Description	Required
`discoveredAppliances`	An array of structures that represents the discoverable devices associated with a customer's device cloud account. If there are no devices associated with the customer account, this property should contain an empty array. The property can be null if an error occurs. The maximum number of items allowed in the array is 300. Properties for each item in the array are listed below	Yes

	must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. The identifier cannot exceed 256 characters.	
`discoveredAppliance.manufacturerName`	The name of the device manufacturer. This value cannot exceed 128 characters.	Yes
`discoveredAppliance.modelName`	Device model name. This value cannot exceed 128 characters.	Yes
`discoveredAppliance.version`	The vendor-provided version of the device. This value cannot exceed 128 characters.	Yes
`discoveredAppliance.friendlyName`	The name used by the customer to identify the device. This value cannot exceed 128 characters and should not contain special characters or punctuation.	Yes
`discoveredAppliance.friendlyDescription`	A human-readable description of the device. This value cannot exceed 128 characters. The description should contain a description of how the device is connected. For example, "WiFi Thermostat connected via Wink".	Yes
`discoveredAppliance.isReachable`	**true** to indicate the device is currently reachable; otherwise, **false**.	Yes

supports. Valid actions for this directive include:

- decrementPercentage
- decrementColorTemperature
- decrementTargetTemperatur e
- getLockState
- getTargetTemperature
- getTemperatureReading
- incrementPercentage
- incrementColorTemperature
- incrementTargetTemperatur e
- setColor
- setColorTemperature
- setLockState
- setPercentage
- setTargetTemperature
- turnOff
- turnOn

	additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.

`DiscoverAppliancesResponse` example:


```
{
"header": {
    "messageId": "ff746d98-ab02-4c9e-9d0d-b44711658414",
    "name": "DiscoverAppliancesResponse",
    "namespace": "Alexa.ConnectedHome.Discovery",
    "payloadVersion": "2"
},
"payload": {
    "discoveredAppliances": [
        {
            "actions": [
                "incrementTargetTemperature",
                "decrementTargetTemperature",
                "getTargetTemperature",
                "setTargetTemperature"
            ],
            "additionalApplianceDetails": {
                "extraDetail1": "optionalDetailForSkillAdapterToReferenceTh
                "extraDetail2": "There can be multiple entries",
                "extraDetail3": "but they should only be used for reference
                "extraDetail4": "This is not a suitable place to maintain c
            },
            "applianceId": "uniqueThermostatDeviceId",
            "friendlyDescription": "descriptionThatIsShownToCustomer",
            "friendlyName": " Bedroom Thermostat",
            "isReachable": true,
            "manufacturerName": "yourManufacturerName",
            "modelName": "fancyThermostat",
            "version": "your software version number here."
        },
        {
            "actions": [
                "incrementPercentage",
                "decrementPercentage",
                "incrementColorTemperature",
                "decrementColorTemperature",
                "setPercentage",
                "setColor",
                "setColorTemperature",
                "turnOn",
                "turnOff"
```

```
                "isReachable": true,
                "manufacturerName": "yourManufacturerName",
                "modelName": "fancyLight",
                "version": "your software version number here."
            }
        ]
    }
}
```

Return To Top

On/Off Messages

The message types turn a target device on or off. These messages are typically used by several different types of devices.

- TurnOnRequest
- TurnOnConfirmation
- TurnOffRequest
- TurnOffConfirmation

TurnOnRequest

Example Utterances:

“Alexa, turn on the *device name*”

“Alexa, schalte *Gerätename* ein”

Purpose: Request to turn on the specified device. Sent from the Smart Home Skill API to the skill adapter.

Property	Value
name	TurnOnRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceID`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

		pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.

`TurnOnRequest` example:

```
{
    "header": {
        "messageId": "01ebf625-0b89-4c4d-b3aa-32340e894688",
        "name": "TurnOnRequest",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        "accessToken": "[OAuth token here]",
        "appliance": {
            "additionalApplianceDetails": {},
            "applianceId": "[Device ID for Ceiling Fan]"
        }
    }
}
```

Return To Top

TurnOnConfirmation

Example Alexa Response: “OK”

Purpose: Indicates the device was successfully turned on. The expected response from the skill adapter to the Smart Home Skill API for a successful `TurnOnRequest`.

Property	Value
name	TurnOnConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`TurnOnConfirmation` example:

```
{
    "header": {
        "messageId": "26fa11a8-accb-4f66-a272-8b1ff7abd722",
        "name": "TurnOnConfirmation",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {}
}
```

Return To Top

TurnOffRequest

Example Utterances:

“Alexa, turn off the *device name*”

“Alexa, schalte *Gerätename* aus”

Purpose: Request to turn off the specified device. Sent from the Smart Home Skill API to the skill adapter.

Header

Property	Value
name	TurnOffRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceID`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

		pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.

`TurnOffRequest` example:

```
{
    "header": {
        "messageId": "01ebf625-0b89-4c4d-b3aa-32340e894688",
        "name": "TurnOffRequest",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        "accessToken": "[OAuth token here]",
        "appliance": {
            "additionalApplianceDetails": {},
            "applianceId": "[Device ID for Ceiling Fan]"
        }
    }
}
```

Return To Top

TurnOffConfirmation

Example Alexa Response: “OK”

Purpose: Indicates the device was successfully turned off. The expected response from the skill adapter to the Smart Home Skill API for a successful `TurnOffRequest`.

Property	Value
name	TurnOffConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`TurnOffConfirmation` example:

```
{
    "header": {
        "messageId": "26fa11a8-accb-4f66-a272-8b1ff7abd722",
        "name": "TurnOffConfirmation",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {}
}
```

Return To Top

Tunable Lighting Control Messages

These messages set the color for a light, or set, increment and decrement the color temperature of a tunable white light. Currently these directives are supported in the US only.

The following directives request and respond to changes to the hue of color-capable light(s). Lighting colors are measured in hue, saturation,

- SetColorConfirmation * US only

The following directives request and respond to changes to the color temperature of tunable white light(s). A tunable white light is a light that you can adjust the 'temperature' of shades of white. Color temperature values are measured in Kelvin. The Smart Home Skill API supports values between 2200K (warm white) to 7000K (cool white), with 2700K being equivalent to a soft white compact fluorescent or LED light.

- SetColorTemperatureRequest * US only
- SetColorTemperatureConfirmation * US only
- IncrementColorTemperatureRequest * US only
- IncrementColorTemperatureConfirmation * US only
- DecrementColorTemperatureRequest * US only
- DecrementColorTemperatureConfirmation * US only

It's important to note that an increment color temperature request is the directive sent when a customer requests an increase in the coolness of a light. This results in a higher color temperature value. The inverse is also true; a request to decrement the color temperature is the directive sent when a customer asks for warmer light, which results in a lower color temperature value.

SetColorRequest

Example Utterances:

"Alexa, set the *device name* to *color*"

"Alexa, set the bedroom light to red"

"Alexa, change the kitchen to the color blue"

from the Smart Home Skill API to the skill adapter.

Header

Property	Value
name	SetColorRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes

`d`	identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	
`appliance.additionalApplianceDetails`	A list of string name-value pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	Yes, but the list can be empty.
`color` object	Describes the color to set for the light. Specified in the Hue, Saturation, Brightness (HSB) color model.	Yes
`color.hue`	A double that indicates the desired hue setting. Valid range is 0.00 to 360.00, inclusive.	Yes

	the desired saturation setting. Valid range is 0.0000 to 1.0000, inclusive.	
`color.brightness`	A double that indicates the desired brightness setting. Valid range is 0.0000 to 1.0000, inclusive.	Yes

`SetColorRequest` example:

```
{
  "header": {
    "messageId": "ABC-123-DEF-456",
    "namespace": "Alexa.ConnectedHome.Control",
    "name": "SetColorRequest",
    "payloadVersion": "2"
  },
  "payload": {
    "accessToken": "[OAuth Token here]",
    "appliance": {
      "applianceId": "[Device ID for RGB-capable bulb]",
      "additionalApplianceDetails": {}
    },
    "color": {
      "hue": 0.0,
      "saturation": 1.0000,
      "brightness": 1.0000
    }
  }
}
```

Return To Top

SetColorConfirmation

Example Alexa Response: OK

Header

Property	Value
name	SetColorConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`achievedState` object	Indicates the state of the device after the color change. This object is required, but note that if you are unable to query the state of the device, or do not want to incur the additional latency of a query, you can return the values sent in the `SetColorRequest`.	Yes
`achievedState.color` object	Indicates the color of the device after the color change.	Yes
`color.hue`	A double that indicates the hue setting. Valid range is 0.00 to 360.00, inclusive.	Yes

	the saturation setting. Valid range is 0.0000 to 1.0000, inclusive.	
`color.brightness`	A double that indicates the brightness setting. Valid range is 0.0000 to 1.0000, inclusive.	Yes

`SetColorConfirmation` example:


```
{
  "header": {
    "messageId": "ABC-123-DEF-456",
    "namespace": "Alexa.ConnectedHome.Control",
    "name": "SetColorConfirmation",
    "payloadVersion": "2"
  },
  "payload": {
    "achievedState": {
      "color": {
        "hue": 0.0,
         "saturation": 1.0000,
         "brightness": 1.0000
      }
    }
  }
}
```


Return To Top

SetColorTemperatureRequest

Example Utterances:

“Alexa, change the *device name* to *shade of white*”

“Alexa, make the living room warm white”

“Alexa, set the kitchen to daylight”

skill adapter. The following table lists some possible color temperature requests and related values that the Smart Home Skill API would send. If your lighting device does not support the requested value, its recommended that you set your device to the nearest possible value.

Shades of White	`colorTemperature` value (in degrees Kelvin)
warm, warm white	2200
incandescent, soft white	2700
white	4000
daylight, daylight white	5500
cool, cool white	7000

Header

Property	Value
name	SetColorTemperatureRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes

`appliance object`	perform the operation on.	
`appliance.applianceId`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes
`appliance.additionalApplianceDetails`	A list of string name-value pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	Yes, but the list can be empty.
`colorTemperature` object	Describes the color temperature to set for the light. Specified in Kelvin degrees.	Yes

ue			indicates the requested color temperature in Kelvin degrees. Valid range is 1000 to 10000, inclusive

`SetColorTemperatureRequest` example:

```
{
  "header": {
    "messageId": "ABC-123-DEF-456",
    "namespace": "Alexa.ConnectedHome.Control",
    "name": "SetColorTemperatureRequest",
    "payloadVersion": "2"
  },
  "payload": {
    "accessToken": "[OAuth Token here]",
    "appliance": {
      "additionalApplianceDetails": {},
      "applianceId": "[Device ID for white-capable bulb]"
    },
    "colorTemperature": {
      "value": 2700
    }
  }
}
```

Return To Top

SetColorTemperatureConfirmation

Example Alexa Response: OK

Purpose: Indicates the device was successfully changed to the color temperature requested. The expected response to a `SetColorTemperatureRequest`, and sent from the skill adapter to the Smart Home Skill API.

name	SetColorTemperatureConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`achievedState` object	Indicates the state of the device after the color change. This object is required, but note that if you are unable to query the state of the device, or do not want to incur the additional latency of a query, you can return the values sent in the `SetColorTemperatureRequest`.	Yes
`achievedState.colorTemperature` object	Indicates the color temperature of the device after the color change.	Yes
`colorTemperature.value`	An integer that indicates the color temperature setting in Kelvin degrees. Valid range is 1000 to 10000, inclusive.	Yes

`SetColorTemperatureConfirmation` example:

```
    "messageId": "ABC-123-DEF-456",
    "namespace": "Alexa.ConnectedHome.Control",
    "name": "SetColorTemperatureConfirmation",
    "payloadVersion": "2"
  },
  "payload": {
    "achievedState": {
      "colorTemperature": {
        "value": 2700
      }
    }
  }
}
```


Return To Top

IncrementColorTemperatureRequest

Example Utterances:

“Alexa, set the *device name cooler/whiter*”

“Alexa, set the dining room cooler”

“Alexa, make the living room light whiter”

Purpose: Request to set the color temperature of the specified device to a cooler light setting, which means raising the Kelvin temperature value of the device. Sent from the Smart Home Skill API to the skill adapter. This directive does not specify a specific setting, but asks for a cooler/whiter setting relative to the current device setting. You can choose how much to adjust the setting of the specified device depending on its current setting, capabilities and other values. If the device cannot be incremented further, set it to its maximum color temperature. If you receive this directive and the specified device is currently set to a color, you should return a `NotSupportedInCurrentModeError` with `currentDeviceMode` set to `COLOR`.

Header

Property	Value
name	IncrementColorTemperatureRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceId`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

`ApplianceDetails`	value pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	empty.

`IncrementColorTemperatureRequest` example:

```
{
  "header": {
    "messageId": "ABC-123-DEF-456",
    "name": "IncrementColorTemperatureRequest",
    "namespace": "Alexa.ConnectedHome.Control",
    "payloadVersion": "2"
  },
  "payload": {
    "accessToken": "[OAuth Token here]",
    "appliance": {
      "additionalApplianceDetails": {},
      "applianceId": "[Device ID for tunable white-capable bulb]",
    }
  }
}
```

Return To Top

IncrementColorTemperatureConfirmation

Example Alexa Response: OK

Purpose: Indicates the device successfully increased its color temperature, meaning the coolness of the light was increased. The

Header

Property	Value
name	IncrementColorTemperatureConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`achievedState` object	Indicates the state of the device after the increase in color temperature.	Yes
`achievedState.colorTemperature` object	Indicates the color temperature of the device after the increase.	Yes
`colorTemperature.value`	An integer that indicates the color temperature setting after the increase, in Kelvin degrees. Valid range is 1000 to 10000, inclusive.	Yes

`IncrementColorTemperature Confirmation` example:

```
{
  "header": {
```

```
  "payload": {
    "achievedState": {
      "colorTemperature": {
        "value": 2700
      }
    }
  }
}
```

Return To Top

DecrementColorTemperatureRequest

Example Utterances:

“Alexa, set the *device name warmer/softer*”

“Alexa, set the dining room softer”

“Alexa, make the living room warmer”

Purpose: Request to set the color temperature of the specified device to a warmer light setting, which means lowering the Kelvin temperature value of the device. Sent from the Smart Home Skill API to the skill adapter. This directive does not specify a specific setting, but asks for a warmer/softer setting relative to the current device setting. You can choose how much to adjust the setting of the specified device depending on its current setting, capabilities and other values. If the device cannot be decremented further, set it to its minimum color temperature. If you receive this directive and the specified device is currently set to a color, you should return a `NotSupportedInCurrentModeError` with the `currentDeviceMode` set to `COLOR`.

Header

Property	Value
name	DecrementColorTemperatureRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceId`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

`ApplianceDetails`	value pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	empty.

`DecrementColorTemperatureRequest` example:

```
{
  "header": {
    "messageId": "ABC-123-DEF-456",
    "name": "DecrementColorTemperatureRequest",
    "namespace": "Alexa.ConnectedHome.Control",
    "payloadVersion": "2"
  },
  "payload": {
    "accessToken": "[OAuth Token here]",
    "appliance": {
      "additionalApplianceDetails": {},
      "applianceId": "[Device ID for tunable white-capable bulb]",
    }
  }
}
```

Return To Top

DecrementColorTemperatureConfirmation

Example Alexa Response: OK

Purpose: Indicates the device successfully decreased its color temperature, meaning warmness of the light was increased. The expected

Header

Property	Value
name	DecrementColorTemperatureConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`achievedState` object	Indicates the state of the device after the decrease in color temperature.	Yes
`achievedState.colorTemperature` object	Indicates the color temperature of the device after the decrease.	Yes
`colorTemperature.value`	An integer that indicates the color temperature setting after the decrease, in Kelvin degrees. Valid range is 1000 to 10000, inclusive.	Yes

`DecrementColorTemperature Confirmation` example:

```
{
  "header": {
```

```
  "payload": {
    "achievedState": {
      "colorTemperature": {
        "value": 2700
      }
    }
  }
}
```

Return To Top

Door Lock Control and Query Messages

These messages enable you to query for the current state and change the state of a lock. Currently these messages are supported in the US only.

- GetLockStateRequest * US only
- GetLockStateResponse * US only
- SetLockStateRequest * US only
- SetLockStateConfirmation * US only

GetLockStateRequest

Example Utterances:

“Alexa, is *lock name locked/unlocked*?”

Purpose: Requests the lock-state for the specified appliance. Sent from the Smart Home Skill API to the skill adapter. Currently supported in the US only.

Header

Property	Value
name	GetLockStateRequest
namespace	Alexa.ConnectedHome.Query

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceId`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

		pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.

`GetLockStateRequest` example:


```
{
   "header":{
       "messageId":"01ebf625-0b89-4c4d-b3aa-32340e894688",
       "name":"GetLockStateRequest",
       "namespace":"Alexa.ConnectedHome.Query",
       "payloadVersion":"2"
   },
   "payload":{
       "accessToken":"[OAuth Token here]",
       "appliance":{
           "applianceId":"[Device ID for front door lock appliance]",
           "additionalApplianceDetails":{
               "extraDetail1":"optionalDetailForSkillAdapterToReferenceThisDev
               "extraDetail2":"There can be multiple entries",
               "extraDetail3":"but they should only be used for reference purp
               "extraDetail4":"This is not a suitable place to maintain curren
           }
       }
   }
}
```


Return To Top

GetLockStateResponse

Example Alexa Reponse: “The front door is locked”

Smart Home Skill API. Currently supported in the US only.

Header

Property	Value
name	GetLockStateResponse
namespace	Alexa.ConnectedHome.Query

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`lockState`	Indicates the locked state of the specified appliance. Valid values are LOCKED, UNLOCKED	Yes

	representing when the lockState above was last retrieved from the target appliance. This helps indicate the freshness of the response, which could affect Alexa's response. Accuracy of this value is device-specific and could be estimated by the skill adapter. Valid values are a standard ISO 8601 format, in UTC with 1 second precision. The RFC 3399 variant is preferred, but negative offsets are not allowed. For example, YYYY-MM-DDThh:mm:ssZ	

`GetLockStateResponse` example:

```
{
    "header":{
        "messageId":"[UUID for message, in canonical hexadecimal format]",
        "name":"GetLockStateResponse",
        "namespace":"Alexa.ConnectedHome.Query",
        "payloadVersion":"2"
    },
    "payload":{
        "lockState":"LOCKED",
        "applianceResponseTimestamp":"2017-01-12T23:20:50.52Z"
    }
}
```

Return To Top

SetLockStateRequest

Example Utterances:

“Alexa, lock the *lock name*”

Header

Property	Value
name	SetLockStateRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceID`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

	pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.
`lockState`	Indicates the requested lock-state of the specified appliance. Valid value for this request is LOCKED.	Yes

`SetLockStateRequest` example:


```
{
    "header":{
        "messageId":"01ebf625-0b89-4c4d-b3aa-32340e894688",
        "name":"SetLockStateRequest",
        "namespace":"Alexa.ConnectedHome.Control",
        "payloadVersion":"2"
    },
    "payload":{
        "accessToken":"[OAuth Token here]",
        "appliance":{
            "applianceId":"[Device ID for front door lock appliance]",
            "additionalApplianceDetails":{
                "extraDetail1":"optionalDetailForSkillAdapterToReferenceThisDev
                "extraDetail2":"There can be multiple entries",
                "extraDetail3":"but they should only be used for reference purp
                "extraDetail4":"This is not a suitable place to maintain curren
            }
        },
        "lockState":"LOCKED"
    }
}
```


Return To Top

Example Alexa Response: "The *front door* is now locked"

Purpose: Indicates the locked state for the specified appliance. The returned lock state value should match the lock state requested in the SetLockStateRequest; otherwise an error has occured. Sent from the skill adapter to the Smart Home Skill API. Currently supported in the US only.

Header

Property	Value
name	GetLockStateResponse
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`lockState`	Indicates the locked state of the specified appliance. Valid value for this directive is LOCKED or UNLOCKED.	Yes

`SetLockStateConfirmation` example:

```
    "header":{
        "messageId":"01ebf625-0b89-4c4d-b3aa-32340e894688",
        "name":"SetLockStateConfirmation",
        "namespace":"Alexa.ConnectedHome.Control",
        "payloadVersion":"2"
    },
    "payload":{
        "lockState":"LOCKED"
    }

}
```

Temperature Control and Query Messages

These message types contain temperatures and provide directives for getting and setting the temperature of a device. It is important to note that Smart Home Skill API always passes temperature settings to your skill adapter in Celsius, and your skill adapter should return temperatures in Celsius. However, customers have the option of toggling between Celsius and Fahrenheit with the Metric On/Off option in the Alexa app, and devices may only accept temperature requests in Celsius or Fahrenheit. This means that you may need to do some conversion between the two temperature scales in your adapter code. You also need to store the temperature to at least two decimal places. This insures enough precision that if temperatures are converted between scales, customers' intents are achieved.

- GetTemperatureReadingRequest * US only
- GetTemperatureReadingResponse * US only
- GetTargetTemperatureRequest * US only
- GetTargetTemperatureResponse * US only
- SetTargetTemperatureRequest
- SetTargetTemperatureConfirmation
- IncrementTargetTemperatureRequest
- IncrementTargetTemperatureConfirmation
- DecrementTargetTemperatureRequest
- DecrementTargetTemperatureConfirmation

GetTemperatureReadingRequest

Example Utterances: "Alexa, what is the temperature of *device name*?"

Header

Property	Value
name	GetTemperatureReadingRequest
namespace	Alexa.ConnectedHome.Query

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceID`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

		pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.

`GetTemperatureReadingRequest` example:

```
{
        "header": {
            "messageId": "[UUID for message, in canonical hexadecimal format]",
            "name": "GetTemperatureReadingRequest",
            "namespace": "Alexa.ConnectedHome.Query",
            "payloadVersion": "2"
        },
        "payload": {
            "accessToken": "[OAuth token here]",
            "appliance": {
                "applianceId": "[Device ID for the specified thermostat, thermc
                "additionalApplianceDetails": {
                    "extraDetail1": "optionalDetailForSkillAdapterToReferenceTh
                    "extraDetail2": "There can be multiple entries",
                    "extraDetail3": "but they should only be used for reference
                    "extraDetail4": "Not a suitable place to maintain current d
                }
            },
        }
    }
}
```

Return To Top

GetTemperatureReadingResponse

Example Alexa Response: “According to *device name*, it’s 70 degrees”

only.

Header

Property	Value
name	GetTemperatureReadingResponse
namespace	Alexa.ConnectedHome.Query

Payload

Property	Description	Required
`temperatureReading object`	Indicates the temperature reading from the specified appliance, in degrees Celsius.	Yes
`temperatureReading.value`	Floating point number that indicates the temperature in degrees Celcius.	Yes

	temperatureReading was last retrieved from the target device. This helps indicate the freshness of the response, which could affect Alexa's response. Accuracy of this value is device-specific and could be estimated by the skill adapter. Valid values are a standard ISO 8601 format, in UTC with 1 second precision. The RFC 3399 variant, without offsets, is preferred. For example, YYYY-MM-DDThh:mm:ssZ	

`GetTemperatureReadingResponse` example:

```
{
    "header": {
        "messageId": "[UUID for message, in canonical hexadecimal format]",
        "name": "GetTemperatureReadingResponse",
        "namespace": "Alexa.ConnectedHome.Query",
        "payloadVersion": "2"
    },
    "payload": {
        "temperatureReading": {
            "value": 21.11
        },
        "applianceResponseTimestamp": "2017-01-12T23:20:50.52Z"
    }
}
```

Return To Top

GetTargetTemperatureRequest

Example Utterances: “Alexa, what is the *device name* set to?

Purpose: Requests the current set temperature (setpoint) of the specified device. Sent from the Smart Home Skill API to the skill adapter. Currently supported in the US only.

Property	Value
name	GetTargetTemperatureRequest
namespace	Alexa.ConnectedHome.Query

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceID`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

		pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.

`GetTargetTemperatureRequest` example:

```
{
    "header": {
        "messageId": "[UUID for message, in canonical hexadecimal format]",
        "name": "GetTargetTemperatureRequest",
        "namespace": "Alexa.ConnectedHome.Query",
        "payloadVersion": "2"
    },
    "payload": {
        "accessToken": "[OAuth token here]",
        "appliance": {
            "applianceId": "[Device ID for the specified thermostat]",
            "additionalApplianceDetails": {
                "extraDetail1": "optionalDetailForSkillAdapterToReferenceThisDe
                "extraDetail2": "There can be multiple entries",
                "extraDetail3": "but they should only be used for reference pur
                "extraDetail4": "This is not a suitable place to maintain curre
            }
        }
    }
}
```

Return To Top

GetTargetTemperatureResponse

Example Alexa Response: “The heat is set to 72 degrees”

Home Skill API. Currently supported in the US only.

Header

Property	Value
name	GetTargetTemperatureResponse
namespace	Alexa.ConnectedHome.Query

Payload

Property	Description	Required
`targetTemperature object`	Indicates the target temperature set by the device in single-setpoint mode in degrees Celsius.	No
`temperatureReading.value`	Floating point number that indicates the target temperature in degrees Celcius.	Yes, when temperatureReading.value is present.
`coolingTargetTemperature` object	Indicates the target temperature (setpoint) for cooling, in degrees Celcius, when a device has dual setpoints. Usually combined with heatingTargetTemperature object.	No

[illegible]	indicates the target temperature for cooling, in degrees Celcius.	coolingTargetTemperature present.
`heatingTargetTemperature` object	Indicates the target temperature (setpoint) for heating, in degrees Celcius, when a device has dual setpoints. Usually combined with coolingTargetTemperature object.	No
`heatingTargetTemperature.value`	Floating point number that indicates the target temperature (setpoint) for heating, in degrees Celcius.	Yes, when heatingTargetTemperature present.
`temperatureMode` object	Indicates the canonical temperature mode set by the device	Yes

temperature mode set by the device. Valid values are:

- AUTO: Indicates automatic heat/cool selection
- COOL: Indicates Cooling mode
- HEAT: Indicates heating mode
- ECO: Indicates economical mode
- OFF: Indicates heating/cooling is turned off, although device may still have power
- CUSTOM: Indicates a custom mode that is specified by `friendlyName`

	name for a temperature mode. Use this value if the device-specific mode and the canonical mode represent the same behavior but are named differently. For example, a device may call a dual mode "Heat Cool", but canonically this is referred to as AUTO. `friendlyName` is set to an empty string, Alexa will not reply with a temperature mode regardless of other values that are returned.	`temperatureMode.value` is set to CUSTOM.
`applianceResponseTimestamp`	Indicates when the target temperature values were was last retrieved from the target device. This helps indicate the freshness of the response, which could affect Alexa's response. Accuracy of this value is device-specific and could be estimated by the skill adapter. Valid values are a standard ISO 8601 format, in UTC with 1 second precision. The RFC 3399 variant, without offsets, is preferred. For example, YYYY-MM-DDThh:mm:ssZ	No

`GetTargetTemperatureResponse` example with a single setpoint:

```
{
    "header": {
        "messageId": "[UUID for message, in canonical hexadecimal format]",
        "name": "GetTargetTemperatureResponse",
        "namespace": "Alexa.ConnectedHome.Query",
        "payloadVersion": "2"
    },
```

```
        "temperatureMode": {
            "value": "HEAT",
            "friendlyName": "Optional device-specific temperature mode name"
        }
    }
}
```

`GetTargetTemperatureResponse` example with a dual setpoint:

```
{
    "header": {
        "messageId": "[UUID for message, in canonical hexadecimal format]",
        "name": "GetTargetTemperaturesResponse",
        "namespace": "Alexa.ConnectedHome.Query",
        "payloadVersion": "2"
    },
    "payload": {
        "coolingTargetTemperature": {
            "value": 23.89,         },
        "heatingTargetTemperature": {
            "value": 22.20,
        },
        "applianceResponseTimestamp": "2017-01-12T23:20:50.52Z"
        "temperatureMode": {
            "value": "AUTO",
            "friendlyName": "Heat-Cool"
        }
    }
}
```

`GetTargetTemperatureResponse` example with a single setpoint and a custom temperature mode:

```
{
    "header": {
        "messageId": "[UUID for message, in canonical hexadecimal format]",
        "name": "GetTargetTemperatureResponse",
        "namespace": "Alexa.ConnectedHome.Query",
        "payloadVersion": "2"
    },
    "payload": {
        "targetTemperature": {
            "value": 20.50
        },
        "applianceResponseTimestamp": "2017-01-12T23:20:50.52Z"
        "temperatureMode": {
            "value": "CUSTOM",
```

`GetTargetTemperatureResponse` example when temperatureMode is OFF.

```
{
    "header": {
        "messageId": "[UUID for message, in canonical hexadecimal format]",
        "name": "GetTargetTemperatureResponse",
        "namespace": "Alexa.ConnectedHome.Query",
        "payloadVersion": "2"
    },
    "payload": {
        "applianceResponseTimestamp": "2017-01-12T23:20:50.52Z"
        "temperatureMode": {
            "value": "OFF",
            "friendlyName": "Optional device-specific temperature mode name"
        }
    }
}
```

Return To Top

SetTargetTemperatureRequest

Example Utterances:

“Alexa, set the *room name* to *number* degrees”

“Alexa, stelle *Raumname* auf *Anzahl* Grad”

Purpose: Requests the specified room name to be set to the specified temperature, in degrees Celsius. Sent from the Smart Home Skill API to the skill adapter.

Header

Property	Value

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceID`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This value cannot exceed 256 characters.	Yes

	pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.
`targetTemperature`	Specifies the target temperature, in degrees Celsius, for the device specified by `applianceId`. Contains a single property, `value`, which specifies a number.	Yes

`SetTargetTemperatureRequest` example:

```
{
    "header": {
        "messageId": "b6602211-b4b3-4960-b063-f7e3967c00c4",
        "name": "SetTargetTemperatureRequest",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        "accessToken": "[OAuth token here]",
        "appliance": {
            "additionalApplianceDetails": {
                "extraDetail1": "optionalDetailForSkillAdapterToReferenceTh
                "extraDetail2": "There can be multiple entries",
                "extraDetail3": "but they should only be used for reference
                "extraDetail4": "This is not a suitable place to maintain c
            },
            "applianceId": "[Device ID for Living Room Thermostat]"
        },
        "targetTemperature": {
            "value": 25.0
        }
    }
}
```

SetTargetTemperatureConfirmation

Example Alexa Response: The *room name* heat is set to *number* degrees

Purpose: Indicates the target temperature was set successfully. The expected response to a `SetTargetTemperatureRequest`, and sent from the skill adapter to the Smart Home Skill API.

Header

Property	Value
name	SetTargetTemperatureConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`targetTemperature`	An object that indicates the target temperature set by the device, in degrees Celsius. Contains a single property, `value`, which specifies a number.	Yes
`temperatureMode`	A object that indicates the temperature mode set by the device. Contains a single property `value` set to one of the following strings: AUTO, COOL, HEAT.	Yes

previousState object	and temperature for the device.	
previousState.targetTemperature	An object that indicates the target temperature set by the device, in degrees Celsius. Contains a single property, value, which specifies a number.	Yes
previousState.mode	A object that indicates the previous mode set by the device. Contains a single property value set to one of the following strings: AUTO, COOL, HEAT.	Yes

SetTargetTemperatureConfirmation example:

```
{
    "header":{
          "namespace":"Alexa.ConnectedHome.Control",
            "name":"SetTargetTemperatureConfirmation",
            "payloadVersion":"2",
            "messageId":"cc36e80c-6357-41e0-9dd4-b76cb3a394e3"
        },
    "payload":{
        "targetTemperature":{
            "value":25.0
        },
    "temperatureMode":{
        "value":"AUTO"
    },
    "previousState":{
        "targetTemperature":{
            "value":21.0
         },
        "mode":{
            "value":"AUTO"
        }
      }
    }
}
```

Return To Top

Example Utterances:

“Alexa, increase the *device name* by *number* degrees”

“Alexa, erhöhe *Gerätename* um *Anzahl* Grad”

Purpose: Requests the temperature of the specified room/device is raised by the specified amount, in degrees Celsius. Sent from the Smart Home Skill API to the skill adapter.

Header

Property	Value
name	IncrementTargetTemperatureRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes

	The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: &. This identifier cannot exceed 256 characters.	
`appliance.additionalApplianceDetails`	A list of string name-value pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	Yes, but the list can be empty.

		indicates the amount to decrease the target temperature of the device by, in degrees Celsius. Contains a single property, `value`, which specifies a number.	deltaTemperature or targetTemperature must be present.
`targetTemperature` Deprecated. Use `deltaTemperature`.	Deprecated.	Indicates the target temperature to set for the device, in degrees Celsius. Contains a single property, `value`, which specifies a number.	Either deltaTemperature or targetTemperature must be present.

`IncrementTargetTemperatureRequest` example:

```
{
    "header": {
        "messageId": "77ff65eb-a015-4777-99ba-6e90d200dd6c",
        "name": "IncrementTargetTemperatureRequest",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        " deltaTemperature": {
            "value": 3.6
        },
        "accessToken": "[OAuth token here]",
        "appliance": {
            "additionalApplianceDetails": {
                "extraDetail1": "optionalDetailForSkillAdapterToReferenceTh
                "extraDetail2": "There can be multiple entries",
                "extraDetail3": "but they should only be used for reference
                "extraDetail4": "This is not a suitable place to maintain c
            },
            "applianceId": "[Device ID for Bedroom Thermostat]"
```

Return To Top

IncrementTargetTemperatureConfirmation

Example Alexa Response: “OK”

Purpose: Indicates that the target temperature for the device was incremented successfully. It is the expected response to an `IncrementTargetTemperatureRequest`, and is sent from the skill adapter to the Smart Home Skill API.

Header

Property	Value
name	IncrementTargetTemperatureConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`targetTemperature`	Indicates the target temperature set by the device, in degrees Celsius. Contains a single property, `value`, which specifies a number.	Yes

	temperature mode set by the device. Contains a single property `value` set to one of the following strings: AUTO, COOL, HEAT.	
`previousState` object	Indicates the temperature and mode before changes were made.	Yes
`previousState.targetTemperature`	Indicates the target temperature set by the device, in degrees Celsius. Contains a single property, `value`, which specifies a number.	Yes
`previousState.mode`	A object that indicates the previous mode set by the device. Contains a single property `value` set to one of the following strings: AUTO, COOL, HEAT.	Yes

`IncrementTargetTemperatureConfirmation` example:


```
{
    "header": {
        "messageId": "780013dd-99d0-4c69-9e35-db0457f9f2a7",
        "name": "IncrementTargetTemperatureConfirmation",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        "previousState": {
            "mode": {
                "value": "AUTO"
            },
            "targetTemperature": {
                "value": 21.0
            }
        },
        "targetTemperature": {
            "value": 25.0
        },
        "temperatureMode": {
```

Return To Top

DecrementTargetTemperatureRequest

Example Utterances:

“Alexa, decrease *device name* by *number* degrees”

“Alexa, reduziere *Gerätename* um *Anzahl* Grad”

Purpose: Requests the temperature of the specified room/device is lowered by the specified amount, in degrees Celsius. Sent from The Smart Home Skill API to the skill adapter.

Header

Property	Value
name	DecrementTargetTemperatureRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required

	the customer's device cloud account.	
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceId`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This identifier cannot exceed 256 characters.	Yes

	additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	
`deltaTemperature`	The amount to decrease the target temperature of the device by, in degrees Celsius. Contains a single property, `value`, which specifies a number.	Either deltaTemperature or targetTemperature must be present.
`targetTemperature` Deprecated. Use `deltaTemperature`.	Deprecated. Indicates the target temperature to set for the device, in degrees Celsius.	Either deltaTemperature or targetTemperature must be present.

`DecrementTargetTemperatureRequest` example:

```
{
  "header": {
    "namespace": "Alexa.ConnectedHome.Control",
    "name": "DecrementTargetTemperatureRequest",
    "payloadVersion": "2",
```

```
      "applianceId": "[Device ID for Bedroom Thermostat]",
      "additionalApplianceDetails": {
        "extraDetail1": "optionalDetailForSkillAdapterToReferenceThisDevice
        "extraDetail2": "There can be multiple entries",
        "extraDetail3": "but they should only be used for reference purpose
        "extraDetail4": "This is not a suitable place to maintain current d
      }
    },
    " deltaTemperature": {
      "value": 1
    }
  }
}
```


Return To Top

DecrementTargetTemperatureConfirmation

Example Alexa Response: “OK”

Purpose: Indicates that the target temperature for the device was decreased successfully. It is the expected response to an `DecrementTargetTemperatureRequest`, and is sent from the skill adapter to the Smart Home Skill API.

Header

Property	Value
name	DecrementTargetTemperatureConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

	temperature set by the device, in degrees Celsius. Contains a single property, `value`, which specifies a number.	
`temperatureMode`	A object that indicates the temperature mode set by the device. Contains a single property `value` set to one of the following strings: AUTO, COOL, HEAT.	Yes
`previousState` object	Indicates the temperature and mode before changes were made.	Yes
`previousState.targetTemperature`	Indicates the target temperature set by the device, in degrees Celsius. Contains a single property, `value`, which specifies a number.	Yes
`previousState.mode`	A object that indicates the previous mode set by the device. Contains a single property `value` set to one of the following strings: AUTO, COOL, HEAT.	Yes

`DecrementTargetTemperatureConfirmation` example:

```
{
    "header": {
        "messageId": "8fab15be-c75a-4d49-b9c3-2dacc24b4c23",
        "name": "DecrementTargetTemperatureConfirmation",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        "previousState": {
            "mode": {
                "value": "AUTO"
            },
```

```
                "value": 27.0
            },
            "temperatureMode": {
                "value": "AUTO"
            }
        }
    }
```

Return To Top

Percentage Messages

These message types set, increment or decrement a target device by a percentage.

- SetPercentageRequest
- SetPercentageConfirmation
- IncrementPercentageRequest
- IncrementPercentageConfirmation
- DecrementPercentageRequest
- DecrementPercentageConfirmation

SetPercentageRequest

“Alexa, set *name* to *number* percent”

“Alexa, stelle *Geräteame* auf *Anzahl* Prozent”

Purpose: Request to adjust the numerical setting of the specified device by the specified percent. Sent from the Smart Home Skill API to the skill adapter.

Property	Value
name	SetPercentageRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes
`appliance.applianceId`	A device identifier. The identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This identifier cannot exceed 256 characters.	Yes

	pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	the list can be empty.
percentageState	The percent change to apply to the device specified as a 64-bit double value with precision of up to two decimal places. Range is from 0.00 to 100.00, inclusive.	Yes

`SetPercentageRequest` example:

```
{
    "header": {
        "messageId": "95872301-4ff6-4146-b3a4-ae84c760c13e",
        "name": " SetPercentageRequest",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        "accessToken": "[OAuth token here]",
        "appliance": {
            "additionalApplianceDetails": {},
            "applianceId": "[Device ID for Cinema Room Light]"
        },
        "percentageState": {
            "value": 50.0
        }
    }
}
```

Return To Top

Example Alexa Response: “OK”

Purpose: Indicates the device was successfully adjusted by the percentage specified. It is the expected response to a `SetPercentageRequest` and is sent from the skill adapter to the Smart Home Skill API.

Header

Property	Value
name	SetPercentageConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`SetPercentageConfirmation` example:

```
{
    "header": {
        "messageId": "6de52aef-e0ee-43f0-bd66-dc71234209c3",
        "name": "SetPercentageConfirmation",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {}
}
```

Return To Top

Example Utterances:

“Alexa, increase *device name* by *number* percent”

“Alexa, erhöhe *Gerätename* um *Anzahl* Prozent”

Purpose: Request to increase the numerical setting of the specified device by the specified percentage. Sent from the Smart Home Skill API to the skill adapter.

Header

Property	Value
name	IncrementPercentageRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	The appliance to perform the operation on.	Yes

	identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This identifier cannot exceed 256 characters.	
`appliance.additionalApplianceDetails`	A list of string name-value pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes, also the Smart Home Skill API does not understand or use this data.	Yes, but the list can be empty.

	deltaPercentage	The percent increase to apply to the device specified as a 64-bit double. For this directive, deltaPercentage is added to the current percentage setting. For example, if the device is currently set to 40%, a deltaPercentage value of 15 means the device will be set at 55% after the request completes. Range is between 0.00 and 100.00, inclusive. If the deltaPercentage value falls within the allowed range, but exceeds the maximum allowed by the target appliance, a ValueOutOfRangeError is returned and the device setting is not be changed.	Extra: deltaPercentage or percentageState is required.

percentageState Deprecated. Use deltaPercentage	Deprecated. The percent change to apply to the device specified as a 64-bit double value with precision of up to two decimal places. Range is from 0.00 to 100.00, inclusive.		deltaPercentage or percentageState is required.

`IncrementPercentageRequest` example:

```
{
    "header": {
        "messageId": "a0c739b9-4c12-48c9-88c7-fc2e1f051b0b",
        "name": "IncrementPercentageRequest",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {
        "accessToken": "[OAuth token here]",
        "appliance": {
            "additionalApplianceDetails": {},
            "applianceId": "[Device ID for Cinema Room Light]"
        },
        "deltaPercentage": {
            "value": 10.0
        }
    }
}
```

Return To Top

IncrementPercentageConfirmation

Example Alexa Response: “OK”

Purpose: Indicates the device was successfully increased by the percentage specified. It is the expected response to a

Header

Property	Value
name	IncrementPercentageConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`IncrementPercentageConfirmation` example:

```
{
    "header": {
        "messageId": "a0c739b9-4c12-48c9-88c7-fc2e1f051b0b",
        "name": " IncrementPercentageConfirmation",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {}
}
```

Return To Top

DecrementPercentageRequest

Example Utterances:

“Alexa, decrease *device name* by *number* percent”

device by the specified percentage. Sent from the Smart Home Skill API to the skill adapter.

Header

Property	Value
name	DecrementPercentageRequest
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
`accessToken`	Access token associated with the customer's device cloud account.	Yes
`appliance` object	Yes	

	identifier must be unique across all devices owned by an end user within the domain for the skill adapter. In addition, the identifier needs to be consistent across multiple discovery requests for the same device. An identifier can contain any letter or number and the following special characters: _ - = # ; : ? @ &. This identifier cannot exceed 256 characters.	
`appliance.additionalApplianceDetails`	A list of string name-value pairs that provide additional information about a device for use by the skill adapter. The contents of this property cannot exceed 5000 bytes. Also, the Smart Home Skill API does not understand or use this data.	Yes, but the list can be empty.

		apply to the device specified as a 64-bit double. For this directive, deltaPercentage is subtracted from the current percent setting. For example, if the device is currently set to 40%, a deltaPercentage value of 15 means the device will be set at 25% after the request completes. Range is between 0.00 and 100.00, inclusive. If the deltaPercentage value falls within the allowed range, but is lower than the minimum allowed by the target appliance, a ValueOutOfRangeError is returned and the device setting is not be changed.
`percentageState`	Deprecated. Use deltaPercentage	Deprecated. The percent change to apply to the device specified as a 64-bit double value with precision of up to two decimal places. Range is from 0.00 to 100.00, inclusive.

`DecrementPercentageRequest` example:

```
{
    "header": {
        "messageId": "7048c18d-4141-4871-bf0e-da3e54dee3f7",
        "name": " DecrementPercentageRequest",
```

```
            "appliance": {
                "additionalApplianceDetails": {},
                "applianceId": "[Device ID for Cinema Room Light]"
            },
            "deltaPercentage": {
                "value": 20.0
            }
        }
    }
```

Return To Top

DecrementPercentageConfirmation

Example Alexa Response: “OK”

Purpose: Indicates the device was successfully decreased by the percentage specified. It is the expected response to a `DecrementPercentageRequest` and is sent from the skill adapter to the Smart Home Skill API.

Header

Property	Value
name	DecrementPercentageConfirmation
namespace	Alexa.ConnectedHome.Control

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

```
{
    "header": {
        "messageId": "17732fa1-9da7-4b03-a8f3-9e6bdf8374e9",
        "name": "DecrementPercentageConfirmation",
        "namespace": "Alexa.ConnectedHome.Control",
        "payloadVersion": "2"
    },
    "payload": {}
}
```

Return To Top

Health Check Messages

These message types check the availability of the skill adapter

- HealthCheckRequest
- HealthCheckResponse

HealthCheckRequest

Example Utterances: N/A

Purpose: Requests the availability of the skill adapter. These are periodically sent by the Smart Home Skill API to the skill adapter.

Header

Property	Value
name	HealthCheckRequest
namespace	Alexa.ConnectedHome.System

Property	Description	Required
`initiationTimestamp`	A timestamp measured in milliseconds since January 1, 1970, which indicates when the health check was sent.	Yes

`HealthCheckRequest` example

```
{
    "header": {
        "messageId": "243550dc-5f95-4ae4-ad43-4e1e7cb037fd",
        "name": " HealthCheckRequest",
        "namespace": "Alexa.ConnectedHome.System",
        "payloadVersion": "2"
    },
    "payload": {
        "initiationTimestamp": "1435302567000"
    }
}
```

Return To Top

HealthCheckResponse

Example Alexa Response: N/A

Purpose: Indicates a successful or failed health check. The expected response to a `HealthCheckRequest`, and sent from the skill adapter to the Smart Home Skill API.

Header

Property	Value
name	HealthCheckResponse

Payload

Property	Description	Required
`isHealthy`	Indicates whether the skill adapter is online and receiving requests.	Yes
`description`	Non-formatted description of skill adapter state.	Yes

Passed `HealthCheckResponse` example:

```
{
    "header": {
        "messageId": "f9905dc8-b861-4912-bcf7-5b90f62b3a71",
        "name": "HealthCheckResponse",
        "namespace": "Alexa.ConnectedHome.System",
        "payloadVersion": "2"
    },
    "payload": {
        "description": "The system is currently healthy",
        "isHealthy": true
    }
}
```

Failed `HealthCheckResponse` example:

```
{
    "header": {
        "messageId": "f9905dc8-b861-4912-bcf7-5b90f62b3a71",
        "name": "HealthCheckResponse",
        "namespace": "Alexa.ConnectedHome.System",
        "payloadVersion": "2"
    },
    "payload": {
        "description": "The system is currently not healthy",
        "isHealthy": false
    }
}
```

Error Messages

There are different kinds of errors that can occur when the Smart Home Skill API sends a control request to your skill adapter, and your skill adapter should respond with the appropriate error type, and supporting information, if required. However, a smart home skill is not required to return every error type; only errors that are appropriate to the type of failure that occurs.The error types and details are listed in this section. Unless otherwise noted, error messages are not applicable to appliance discovery, and an error message should never be returned as response to a DiscoverAppliancesRequest.

User Faults: These errors occur when the request is invalid due to customer error. For example the customer asks to set a thermostat to 1000 degrees.

- ValueOutOfRangeError
- TargetOfflineError
- NoSuchTargetError
- BridgeOfflineError

Skill Adapter Faults: These errors occur when the request is valid but the skill adapter cannot complete the required task because of a hardware issue or limitation.

- DriverInternalError
- DependentServiceUnavailableError
- NotSupportedInCurrentModeError
- RateLimitExceededError
- TargetBridgeConnectivityUnstableError

- TargetHardwareMalfunctionError
- TargetBridgetHardwareMalfunctionError
- TargetConnectivityUnstableError
- TargetHardwareMalfunctionError
- UnableToGetValueError
- UnableToSetValueError
- UnwillingToSetValueError

Other Faults: These errors occur when the request cannot be fulfilled due to content in the request; either the authentication token is not valid, or some other aspect of the request cannot be fulfilled by the skill adapter.

- ExpiredAccessTokenError
- InvalidAccessTokenError
- UnsupportedTargetError
- UnsupportedOperationError
- UnsupportedTargetSettingError
- UnexpectedInformationReceivedError

User Faults

The following errors occur when a customer gives incorrect instructions or instructions that cannot be completed to Alexa.

ValueOutOfRangeError

Purpose: Indicates a customer request would set a target value to a value out of its supported range. For example, a customer asks, “Alexa, set the kitchen to 1000 degrees”.

Property	Description	Required
`minimumValue`	A 64-bit double indicating the lowest value allowed for the target device setting.	Yes
`maximumValue`	A 64-bit double indicating highest value allowed for the target device setting.	Yes

`ValueOutOfRangeError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":" ValueOutOfRangeError",
    "payloadVersion":"2",
    "messageId":"697fe957-c842-4545-a159-8a8c75fbe5bd"
  },
  "payload":{
    "minimumValue":15.0,
    "maximumValue":30.0
  }
}
```

Return To Top

TargetOfflineError

Purpose: Indicates that the target device is not connected to the customer's device cloud or is not on.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"TargetOfflineError",
    "payloadVersion":"2",
    "messageId":"15a248f6-8ab5-433d-a3ac-73c358e0bebd"
  },
  "payload":{
  }
}
```


Return To Top

NoSuchTargetError

Purpose: Indicates that the target device cannot be found, meaning it was never configured by the end-user.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`NoSuchTargetError` example:


```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"NoSuchTargetError",
    "payloadVersion":"2",
    "messageId":"bc652339-1b09-423d-b679-1bd19ae59245"
  },
  "payload":{
  }
}
```


Return To Top

Purpose: Indicates the target device is connected to a home automation hub or bridge, which is powered off.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`BridgeOfflineError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"BridgeOfflineError",
    "payloadVersion":"2",
    "messageId":"49f72397-858f-41cb-a7d3-b8cfa4c5fd0f"
  },
  "payload":{
  }
}
```

Return To Top

Skill Adapter Faults

The following errors occur when the skill adapter has problems interacting with the cloud-enabled device or device-cloud. In these situations, the customer request is valid, but cannot be completed for some reason.

DriverInternalError

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`DriverInternalError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"DriverInternalError",
    "payloadVersion":"2",
    "messageId":"e1ee71ed-952d-45fa-b2f4-2907649f48dc"
  },
  "payload":{
  }
}
```

Return To Top

DependentServiceUnavailableError

Purpose: Indicates that a skill adapter dependency is unavailable and the skill adapter cannot complete the request.

Payload

Property	Description	Required

	alphanumeric characters and spaces. This value truncates after 256 characters.

DependentServiceUnavailableError example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"DependentServiceUnavailableError",
    "payloadVersion":"2",
    "messageId":"e1929526-66fb-4f99-869a-13c58bee88ef"
  },
  "payload":{
    "dependentServiceName":"Customer Credential Database"
  }
}
```

Return To Top

TargetConnectivityUnstableError

Purpose: Indicates the cloud-connectivity for the target device is not stable and reliable.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

TargetConnectivityUnstableError example:

```
    "payloadVersion":"2",
    "messageId":"502f0076-355c-4a5e-bb15-3ab7d78b8278"
  },
  "payload":{
  }
}
```


Return To Top

TargetBridgeConnectivityUnstableError

Purpose: Indicates that cloud-connectivity for a home automation hub or bridge that connects the target device is unstable and unreliable.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`TargetBridgeConnectivityUnstableError` example:


```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"TargetBridgeConnectivityUnstableError",
    "payloadVersion":"2",
    "messageId":"502f0076-355c-4a5e-bb15-3ab7d78b8278"
  },
  "payload":{
  }
}
```


Return To Top

Purpose: Indicates that the target device has outdated firmware.

Payload

Property	Description	Required
`minimumFirmwareVersion`	Alphanumeric value indicating minimum allowed firmware version. Cannot exceed 256 characters.	Yes
`currentFirmwareVersion`	Alphanumeric value indicating current firmware version. Cannot exceed 256 characters.	Yes

`TargetFirmwareOutdatedError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"TargetFirmwareOutdatedError",
    "payloadVersion":"2",
    "messageId":"917314cd-ca00-49ca-b75e-d6f65ac43503"
  },
  "payload":{
    "minimumFirmwareVersion":"17",
    "currentFirmwareVersion":"6"
  }
}
```

Return To Top

TargetBridgeFirmwareOutdatedError

Payload

Property	Description	Required
`minimumFirmwareVersion`	Alphanumeric value indicating minimum allowed firmware version. Cannot exceed 256 characters.	Yes
`currentFirmwareVersion`	Alphanumeric value indicating current firmware version. Cannot exceed 256 characters.	Yes

TargetBridgeFirmwareOutdatedError example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"TargetBridgeFirmwareOutdatedError",
    "payloadVersion":"2",
    "messageId":"917314cd-ca00-49ca-b75e-d6f65ac43503"
  },
  "payload":{
    "minimumFirmwareVersion":"17",
    "currentFirmwareVersion":"6"
  }
}
```

Return To Top

TargetHardwareMalfunctionError

Purpose: Indicates that the target device experienced a hardware malfunction.

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`TargetHardwareMalfunctionError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"TargetHardwareMalfunctionError",
    "payloadVersion":"2",
    "messageId":"3840d1ad-05fc-413c-b2ad-2aa237090a29"
  },
  "payload":{}
}
```

Return To Top

TargetBridgeHardwareMalfunctionError

Indicates that the home automation hub or bridge connecting the target device experienced a hardware malfunction.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`TargetBridgeHardwareMalfunctionError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
```

```
    }
  }
```

Return To Top

UnableToGetValueError

Purpose: Indicates that an error occurred while trying to get the specified value on the target device. When returning this error, an appropriate `errorInfo.code` value enables Alexa to respond appropriately for different kinds of failures. You only need to generate an error code appropriate for the target device.

Payload

Property	Description	Required
`errorInfo`	An error object describing why the value can't be set.	Yes

	values are • DEVICE_AJAR: Cannot get the specified state because the door is open. • DEVICE_BUSY: The device is busy • DEVICE_JAMMED: The device is jammed. • DEVICE_OVERHEATED: The device has overheated. • HARDWARE_FAILURE: Request failed because of an undetermined hardware failure. • LOW_BATTERY: The device's battery is low • NOT_CALIBRATED: The device is not calibrated.	
`errorInfo.Description`	A custom description of the error from the device manufacturer.	No

`UnableToGetValueError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Query",
    "name":"UnableToGetValueError",
    "payloadVersion":"2",
    "messageId":"917314cd-ca00-49ca-b75e-d6f65ac43503"
  },
  "payload":{
    "errorInfo":{
      "code":"DEVICE_JAMMED",
      "description":"A custom description of the error.."
```

Return To Top

UnableToSetValueError

Purpose: Indicates that an error occurred while trying to set the specified value on the target device. When returning this error, an appropriate `errorInfo.code` value enables Alexa to respond appropriately for different kinds of failures. You only need to generate error codes appropriate for the target device.

Payload

Property	Description	Required
`errorInfo`	An error object describing why the value can't be set.	Yes

	- DEVICE_AJAR - The door or window containing the device is open. - DEVICE_BUSY - The device is busy - DEVICE_JAMMED -The device is jammed. - DEVICE_OVERHEATED - The device has overheated. - HARDWARE_FAILURE - An undetermined hardware failure has occurred. - LOW_BATTERY - The device's battery is low - NOT_CALIBRATED - The device is not calibrated	
errorInfo.description	A custom description of the error from the device manufacturer.	No

`UnableToSetValueError example:

```
{
    "header":{
        "messageId":"3840d1ad-05fc-413c-b2ad-2aa237090a29",
        "name":"UnableToSetValueError",
        "namespace":"Alexa.ConnectedHome.Control",
        "payloadVersion":"2"
    },
    "payload":{
        "errorInfo":{
            "code":"DEVICE_BUSY",
            "description":"A custom description of the error"
        }
    }
```

Return To Top

UnwillingToSetValueError

Purpose: Indicates that the target device partner is unwilling to set the requested value on the specified device. Use this error for temperature settings.

Payload

Property	Description	Required
`errorInfo`	An error object describing why the value can't be set.	Yes
`errorInfo.code`	An error code in string format. Currently, the valid value for code is `ThermostatIsOff`, which indicates the requested operation was rejected because the thermostat is off and the manufacturer is unwilling to automatically turn it on for safety reasons.	Yes
`errorInfo.description`	A custom description of the error from the device manufacturer.	Yes

`UnwillingToSetValueError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"UnwillingToSetValueError",
    "payloadVersion":"2",
    "messageId":"917314cd-ca00-49ca-b75e-d6f65ac43503"
  },
  "payload":{
```

```
}
```

Return To Top

RateLimitExceededError

Purpose: Indicates that the maximum number of requests that a device accepts has been exceeded. This message provides information about the maximum number of requests for a device and the time unit for those requests. For example, if a device accepts four requests per hour, the message should specify 4 and HOUR as *rateLimit* and *timeUnit*, respectively.

Payload

Property	Description	Required
`rateLimit`	An integer that represents the maximum number of requests a device will accept in the specifed time unit.	Yes
`timeUnit`	An all-caps string that indicates the time unit for *rateLimit* such as MINUTE, HOUR or DAY.	Yes

`RateLimitExceededError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"RateLimitExceededError",
```

```
      "timeUnit":"HOUR"
    }
  }
```

Return To Top

NotSupportedInCurrentModeError

Purpose: Indicates that the target device is in a mode in which it cannot be controlled with the Smart Home Skill API, and provides information about the current mode of the device.

For lights, return this error with a `currentDeviceMode` value of `COLOR` to indicate that an `incrementColorTemperature` or `decrementColorTemperature` request was made to a light that is currently set to an HSB color. In this scenario, Alexa responds, “That only works when your light is set to a shade of white.”

Payload

Property	Description	Required
`currentDeviceMode`	A string that represents the current mode of the device. Valid values are AUTO, AWAY, COLOR, COOL, HEAT, and OTHER.	Yes

`NotSupportedInCurrentModeError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"NotSupportedInCurrentModeError",
    "payloadVersion":"2",
```

```
}
```

Return To Top

Other Faults

The following errors occur when one or more of the request inputs is invalid and cannot be handled by the skill adapter. For example, the authentication token is not valid.

ExpiredAccessTokenError

Purpose: Indicates that the access token used for authentication has expired and is no longer valid.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`ExpiredAccessTokenError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"ExpiredAccessTokenError",
    "payloadVersion":"2",
    "messageId":"e1ee71ed-952d-45fa-b2f4-2907649f48dc"
  },
  "payload":{
  }
}
```

InvalidAccessTokenError

Purpose: Indicates that the access token used for authentication is not valid for a reason other than it has expired.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`InvalidAccessTokenError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"InvalidAccessTokenError",
    "payloadVersion":"2",
    "messageId":"e1ee71ed-952d-45fa-b2f4-2907649f48dc"
  },
  "payload":{
  }
}
```

Return To Top

UnsupportedTargetError

Purpose: Indicates that the target device is not supported by the skill adapter.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`UnsupportedTargetError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"UnsupportedTargetError",
    "payloadVersion":"2",
    "messageId":"e1ee71ed-952d-45fa-b2f4-2907649f48dc"
  },
  "payload":{
  }
}
```

Return To Top

UnsupportedOperationError

Purpose: Indicates that the requested operation is not supported on the target device.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`UnsupportedOperationError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
```

```
    }
  }
```

Return To Top

UnsupportedTargetSettingError

Purpose: Indicates that the requested setting is not valid for the specified device and operation.

Payload

Property	Description	Required
None	No required or optional fields in the payload.	N/A

`UnsupportedTargetSettingError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"UnsupportedTargetSettingError",
    "payloadVersion":"2",
    "messageId":"e1ee71ed-952d-45fa-b2f4-2907649f48dc"
  },
  "payload":{
  }
}
```

Return To Top

skill adapter because it was malformed.

Payload

Property	Description	Required
faultingParameter	The property or field in the request message that was malformed or unexpected, and could not be handled by the skill adapter.	Yes

`UnexpectedInformationReceivedError` example:

```
{
  "header":{
    "namespace":"Alexa.ConnectedHome.Control",
    "name":"UnexpectedInformationReceivedError",
    "payloadVersion":"2",
    "messageId":"e1ee71ed-952d-45fa-b2f4-2907649f48dc"
  },
  "payload":{'
    "faultingParameter": "value"
  }
}
```

Return To Top

Changes to Smart Home Skill API Reference - April 7, 2017

- New directives added to control the color and tunable white light settings of devices
  - SetColorRequest
  - SetColorConfirmation

- IncrementColorTemperatureRequest
- IncrementColorTemperatureConfirmation
- DecrementColorTemperatureRequest
- DecrementColorTemperatureConfirmation

- New value added to NotSupportedInCurrentModeError.currentDeviceMode to support lighting changes.

Changes to Smart Home Skill API Reference - February 28, 2017

- New directives added for querying for for lock status and setting lock status.
  - GetLockStateRequest
  - GetLockStateResponse
  - SetLockStateRequest
  - SetLockStateConfirmation
- New error messages specific to lock error conditions.
  - UnableToGetValueError
  - UnableToSetValueError

Changes to Smart Home Skill API Reference - February 17, 2017

- New directives added for querying for temperatures. See:
  - GetTemperatureReadingRequest
  - GetTemperatureReadingResponse
  - GetTargetTemperatureRequest

discoverability.