Alexa SKills Kit SDK for Python Documentation

Release 1.6.0

Alexa Skills Kit

Feb 04, 2019

Guides

The **ASK SDK for Python** makes it easier for you to build highly engaging skills, by allowing you to spend more time on implementing features and less on writing boiler-plate code.

To help you get started with the SDK we have included the following guides. In the future, we plan to include more documentation and samples too.

CHAPTER 1

Guides

1.1 Setting Up The ASK SDK

1.1.1 Introduction

This guide describes how to install the ASK SDK for Python in preparation for developing an Alexa skill.

1.1.2 Prerequisites

The ASK SDK for Python requires **Python 2 (>= 2.7)** or **Python 3 (>= 3.6)**. Before continuing, make sure you have a supported version of Python installed. To show the version, from a command prompt run the following command:

```
$ python --version
Python 3.6.5
```

You can download the latest version of Python here.

1.1.3 Adding the ASK SDK for Python to Your Project

You can download and install the ASK SDK for Python from the Python Package Index (PyPI) using the command line tool `pip`. If you are using **Python 2 version 2.7.9 or later** or **Python 3 version 3.4 or later**, `pip` should be installed with Python by default.

Many Python developers prefer to work in a virtual environment, which is an isolated Python environment that helps manage project dependencies and package versions. The easiest way to get started is to install the SDK in a virtual environment. See the section *Set up the SDK in a virtual environment*.

Another option is to install the ASK SDK for Python to a specific folder. This ensures that you have the required dependencies and makes it easy to locate and deploy the required files for your finished skill. See the section *Set up the SDK in a specific folder*.

Tip: The following steps showcase the installation process for the standard SDK distribution. The standard SDK distribution `ask-sdk` is the easiest way to quickly get up and running with the SDK. It includes the core SDK package, the model package, and the package for the Amazon DynamoDB persistence adapter that enables storing skill attributes in DynamoDB.

If you do not need everything in the standard distribution `ask-sdk`, you can install the core package and expand with individual add-on packages later.

For doing that, change the package name in the `pip install <package name>` commands in the following sections with `ask-sdk-core` for the core package and `ask-sdk-dynamodb-persistence-adapter` for the DynamoDB persistence adapter add-on.

Option 1: Set up the SDK in a virtual environment

This option requires you to install the `virtualenv` package. `virtualenv` is a tool to create isolated Python environments. To get started, from a command prompt, use the following command to install the package:

```
$ pip install virtualenv
```

Next, create a new folder for your Alexa skill and navigate to the folder:

```
$ mkdir skill
$ cd skill
```

Next, create a virtual environment called `skill_env` by issuing the following command:

```
$ virtualenv skill_env
```

Next, activate your virtual environment and install the sdk.

MacOS / Linux

Run the following command to activate your virtual environment:

```
$ source skill_env/bin/activate
```

The command prompt should now be prefixed with *(skill_env)*, indicating that you are working inside the virtual environment. Use the following command to install the ASK Python SDK:

```
(skill_env)$ pip install ask-sdk
```

Depending on the version of Python you are using, the SDK will be installed into the `skill_env/lib/Python3.6/site-packages` folder. The site-packages folder is populated with directories including:

```
ask_sdk
ask_sdk_core
ask_sdk_dynamodb
ask_sdk_model
boto3
...
```

Windows

Run the following command to activate your virtual environment:

```
$ skill_env\Scripts\activate
```

The command prompt should now be prefixed with *(skill_env)*, indicating that you are working inside the virtual environment. Use the following command to install the ASK Python SDK:

```
(skill_env)$ pip install ask-sdk
```

The SDK will be installed into the `skill\Lib\site-packages` folder. The site-packages folder is populated with directories including:

```
ask_sdk
ask_sdk_core
ask_sdk_dynamodb
ask_sdk_model
boto3
...
```

Option 2: Set up the SDK in a specific folder

To get started, from a command prompt create a new folder for your Alexa skill and navigate to the folder:

```
$ mkdir skill
$ cd skill
```

Next, install the ASK SDK for Python using pip. The `-t` option targets a specific folder for installation:

```
$ pip install ask-sdk -t skill_env
```

This creates a folder named `skill_env` inside your `skill` folder and installs the ASK SDK for Python and its dependencies. Your `skill` directory should now contain the folder `skill_env`, which is populated with directories including:

```
ask_sdk
ask_sdk_core
ask_sdk_dynamodb
ask_sdk_model
boto3
...
```

Note: If using Mac OS X and you have Python installed using Homebrew, the preceding command will not work. A simple workaround is to add a `setup.cfg` file in your **ask-sdk** directory with the following content:

```
[install]
prefix=
```

Navigate to the skill_env folder and run the pip install command:

```
$ cd skill_env
$ pip install ask-sdk -t .
```

More on this can be checked on the homebrew docs

1.1.4 Next Steps

Now that you've added the SDK to your project, you're ready to begin developing your skill. Proceed to the next section Developing Your First Skill, for instructions on getting started with a basic skill.

1.2 Developing Your First Skill

The Getting Started guide showed how to set up and install the ASK SDK for Python into a specific directory or into a virtual environment using virtualenv. This guide walks you through developing your first skill with the ASK SDK for Python.

1.2.1 Prerequisites

In addition to an installed version of the ASK SDK for Python you need:

- An Amazon Developer account. This is required to create and configure Alexa skills.
- An Amazon Web Services (AWS) account. This is required for hosting a skill on AWS Lambda.

1.2.2 Creating Hello World

You'll write your Hello World in a single python file named `hello_world.py`. In the `skill` folder that you have created earlier, use your favorite text editor or IDE to create a file named `hello_world.py`.

1.2.3 Implementing Hello World

Request handlers

A custom skill needs to respond to events sent by the Alexa service. For instance, when you ask your Alexa device (e.g. Echo, Echo Dot, Echo Show, etc.) to 'open hello world', your skill needs to respond to the LaunchRequest that is sent to your Hello World skill. With the ASK SDK for Python, you simply need to write a request handler, which is code to handle incoming requests and return a response. Your code is responsible for making sure that the right request handler is used to process incoming requests and for providing a response. The ASK SDK for Python provides two ways to create request handlers:

1. Implement the `AbstractRequestHandler` class under `ask_sdk_core.dispatch_components` package. The class should contain implementations for `can_handle` and `handle` methods. This is described under *Implementation using classes* section.
2. Use the `request_handler` decorator in instantiated skill builder object to tag functions that act as handlers for different incoming requests. This is described under *Implementation using decorators* section.

The implementation of the Hello World skill explores using handler classes first and then shows how to write the same skill using decorators. The functionality of these is identical and you can use **either**.

Exception handlers

Sometimes things go wrong, and your skill code needs a way to handle the problem gracefully. The ASK SDK for Python supports exception handling in a similar way to handling requests. You have a choice of using *classes* or *decorators*. The following implementation sections explore how to implement exception handling.

Tip: You may use either *Implementation using classes* or *Implementation using decorators* options to write a skill.

Warning: We strongly recommend you to choose **one** of the options and use it consistently throughout your skill, for better code structure.

1.2.4 Option 1: Implementation using handler classes

Start by creating a skill builder object. The skill builder object helps in adding components responsible for handling input requests and generating custom responses for your skill.

Type or paste the following code into your `hello_world.py` file.

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()
```

To use handler classes, each request handler is written as a class that implements two methods of the `AbstractRequestHandler` class; `can_handle` and `handle`.

The `can_handle` method returns a Boolean value indicating if the request handler can create an appropriate response for the request. The `can_handle` method has access to the request type and additional attributes that the skill may have set in previous requests or even saved from a previous interaction. The Hello World skill only needs to reference the request information to decide if each handler can respond to an incoming request.

LaunchRequest handler

The following code example shows how to configure a handler to be invoked when the skill receives a LaunchRequest. The LaunchRequest event occurs when the skill is invoked without a specific intent.

Type or paste the following code into your `hello_world.py` file, after the previous code.

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.utils import is_request_type, is_intent_name
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_model import Response
from ask_sdk_model.ui import SimpleCard

class LaunchRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_request_type("LaunchRequest")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speech_text = "Welcome to the Alexa Skills Kit, you can say hello!"

        handler_input.response_builder.speak(speech_text).set_card(
           SimpleCard("Hello World", speech_text)).set_should_end_session(
           False)
        return handler_input.response_builder.response
```

The can_handle function returns **True** if the incoming request is a LaunchRequest. The handle function generates and returns a basic greeting response.

HelloWorldIntent handler

The following code example shows how to configure a handler to be invoked when the skill receives an intent request with the name HelloWorldIntent. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
class HelloWorldIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("HelloWorldIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speech_text = "Hello World"

        handler_input.response_builder.speak(speech_text).set_card(
            SimpleCard("Hello World", speech_text)).set_should_end_session(
            True)
        return handler_input.response_builder.response
```

The can_handle function detects if the incoming request is an IntentRequest, and returns **True** if the intent name is HelloWorldIntent. The handle function generates and returns a basic "Hello World" response.

HelpIntent handler

The following code example shows how to configure a handler to be invoked when the skill receives the built-in intent AMAZON.HelpIntent. Type or paste the following code into your `hello_world.py file`, after the previous handler.

```
class HelpIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("AMAZON.HelpIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speech_text = "You can say hello to me!"

        handler_input.response_builder.speak(speech_text).ask(speech_text).set_card(
            SimpleCard("Hello World", speech_text))
        return handler_input.response_builder.response
```

Similar to the previous handler, this handler matches an IntentRequest with the expected intent name. Basic help instructions are returned, and `.ask(speech_text)` causes the user's microphone to open up for the user to respond.

CancelAndStopIntent handler

The CancelAndStopIntentHandler is similar to the HelpIntent handler, as it is also triggered by the built-In AMAZON.CancelIntent or AMAZON.StopIntent Intents. The following example uses a single handler to respond to both intents. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
class CancelAndStopIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("AMAZON.CancelIntent")(handler_input)
```

(continues on next page)

(continued from previous page)

```
                  or is_intent_name("AMAZON.StopIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speech_text = "Goodbye!"

        handler_input.response_builder.speak(speech_text).set_card(
            SimpleCard("Hello World", speech_text))
        return handler_input.response_builder.response
```

The response to both intents is the same, so having a single handler reduces repetitive code.

SessionEndedRequest handler

Although you cannot return a response with any speech, card or directives after receiving a SessionEndedRequest, the SessionEndedRequestHandler is a good place to put your cleanup logic. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
class SessionEndedRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_request_type("SessionEndedRequest")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        # any cleanup logic goes here

        return handler_input.response_builder.response
```

Implementing exception handlers

The following sample adds a *catch all* exception handler to your skill, to ensure the skill returns a meaningful message for all exceptions. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
from ask_sdk_core.dispatch_components import AbstractExceptionHandler

class AllExceptionHandler(AbstractExceptionHandler):

    def can_handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> bool
        return True

    def handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> Response
        # Log the exception in CloudWatch Logs
        print(exception)

        speech = "Sorry, I didn't get it. Can you please say it again!!"
        handler_input.response_builder.speak(speech).ask(speech)
        return handler_input.response_builder.response
```

Creating the Lambda handler

The Lambda handler is the entry point for your AWS Lambda function. The following code example creates a Lambda handler function to route all inbound requests to your skill. The Lambda handler function creates an SDK skill instance configured with the request handlers that you just created. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
sb.add_request_handler(LaunchRequestHandler())
sb.add_request_handler(HelloWorldIntentHandler())
sb.add_request_handler(HelpIntentHandler())
sb.add_request_handler(CancelAndStopIntentHandler())
sb.add_request_handler(SessionEndedRequestHandler())

sb.add_exception_handler(AllExceptionHandler())

handler = sb.lambda_handler()
```

1.2.5 Option 2: Implementation using decorators

The following code implements the same functionality as above but uses function decorators. You can think of the decorators as a replacement for the `can_handle` method implemented in each handler class above.

Start by creating a skill builder object. The skill builder object helps in adding components responsible for handling input requests and generating custom responses for your skill.

Type or paste the following code into your `hello_world.py` file.

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()
```

LaunchRequest handler

The following code example shows how to configure a handler to be invoked when the skill receives a LaunchRequest. The LaunchRequest event occurs when the skill is invoked without a specific intent.

Type or paste the following code into your `hello_world.py` file, after the previous code.

```
from ask_sdk_core.utils import is_request_type, is_intent_name
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_model import Response
from ask_sdk_model.ui import SimpleCard

@sb.request_handler(can_handle_func=is_request_type("LaunchRequest"))
def launch_request_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "Welcome to the Alexa Skills Kit, you can say hello!"

    handler_input.response_builder.speak(speech_text).set_card(
         SimpleCard("Hello World", speech_text)).set_should_end_session(
         False)
    return handler_input.response_builder.response
```

Similar to the `can_handle` function for the LaunchRequestHandler in the Class pattern, the decorator returns **True** if the incoming request is a LaunchRequest. The `handle` function generates and returns a basic greeting response in the same way the handle function works for the Class pattern.

HelloWorldIntent handler

The following code example shows how to configure a handler to be invoked when the skill receives an intent request with the name HelloWorldIntent. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
@sb.request_handler(can_handle_func=is_intent_name("HelloWorldIntent"))
def hello_world_intent_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "Hello World!"

    handler_input.response_builder.speak(speech_text).set_card(
        SimpleCard("Hello World", speech_text)).set_should_end_session(
        True)
    return handler_input.response_builder.response
```

HelpIntent handler

The following code example shows how to configure a handler to be invoked when the skill receives the built-in intent AMAZON.HelpIntent. Type or paste the following code into your `hello_world.py file`, after the previous handler.

```
@sb.request_handler(can_handle_func=is_intent_name("AMAZON.HelpIntent"))
def help_intent_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "You can say hello to me!"

    handler_input.response_builder.speak(speech_text).ask(speech_text).set_card(
        SimpleCard("Hello World", speech_text))
    return handler_input.response_builder.response
```

Similar to the previous handler, this handler matches an IntentRequest with the expected intent name. Basic help instructions are returned, and `.ask(speech_text)` causes the user's microphone to open up for the user to respond.

CancelAndStopIntent handler

The CancelAndStopIntentHandler is similar to the HelpIntent handler, as it is also triggered by the built-in AMAZON.CancelIntent or AMAZON.StopIntent intents. The following example uses a single handler to respond to both Intents. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
@sb.request_handler(
    can_handle_func=lambda handler_input :
        is_intent_name("AMAZON.CancelIntent")(handler_input) or
        is_intent_name("AMAZON.StopIntent")(handler_input))
def cancel_and_stop_intent_handler(handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "Goodbye!"

    handler_input.response_builder.speak(speech_text).set_card(
        SimpleCard("Hello World", speech_text))
    return handler_input.response_builder.response
```

In the above example, `can_handle` needs a function to be passed. `is_intent_name` returns a function, but we need to check if the request is either *AMAZON.CancelIntent* or *AMAZON.StopIntent*. We achieve this by creating an anonymous function on the fly using Python's in-built `lambda` function.

The response to both intents is the same, so having a single handler reduces repetitive code.

SessionEndedRequest handler

Although you cannot return a response with any speech, card or directives after receiving a SessionEndedRequest, the SessionEndedRequestHandler is a good place to put your cleanup logic. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
@sb.request_handler(can_handle_func=is_request_type("SessionEndedRequest"))
def session_ended_request_handler(handler_input):
    # type: (HandlerInput) -> Response
    # any cleanup logic goes here

    return handler_input.response_builder.response
```

Implementing exception handlers

The following sample adds a *catch all* exception handler to your skill, to ensure the skill returns a meaningful message in case of all exceptions. Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
@sb.exception_handler(can_handle_func=lambda i, e: True)
def all_exception_handler(handler_input, exception):
    # type: (HandlerInput, Exception) -> Response
    # Log the exception in CloudWatch Logs
    print(exception)

    speech = "Sorry, I didn't get it. Can you please say it again!!"
    handler_input.response_builder.speak(speech).ask(speech)
    return handler_input.response_builder.response
```

Creating the Lambda handler

The Lambda handler is the entry point for your AWS Lambda function. The following code example creates a Lambda handler function to route all inbound requests to your skill. The Lambda Handler function creates an SDK skill instance configured with the request handlers that you just created.

Type or paste the following code into your `hello_world.py` file, after the previous handler.

```
handler = sb.lambda_handler()
```

When using decorators, your request handlers and exception handlers are automatically recognized by the Skill Builder object instantiated at the top of the code.

1.2.6 Full source code

The full source code for `hello_world.py` can be found here.

1.2.7 Preparing your code for AWS Lambda

Your code is now complete and you need to create .zip files that contain the files ready to upload to Lambda.

When you upload your code to AWS Lambda, you must include your skill code and its dependencies inside a zip file as a flat file structure, so you'll place your code in the same folder as the ASK SDK for Python, before zipping it.

SDK Setup in Virtual Environment

If you set up the SDK using a virtual environment, the dependencies are installed in the `site-packages` folder in your virtual environment. So, navigate to the `site-packages` folder in `skill_env`.

Note: On **Windows** the `site-packages` folder is located inside the `skill_env\Lib` folder.

Note: For **MacOS/Linux** the `site-packages` folder location depends on the version of Python you are using. For instance *Python 3.6* users will find `site-packages` inside the `skill_env/lib/Python3.6` folder.

Copy the `hello_world.py` file into the `site-packages` folder and create a .zip file of the contents of the folder (**not** the folder itself). Name the file `skill.zip`. You can check the AWS Lambda docs to get more information on creating a deployment package.

SDK Setup in specific folder

If you set up the SDK in a specific folder, the dependencies are installed in that specific folder. That would be `skill_env` folder if you followed the steps mentioned in the getting started guide.

Copy the `hello_world.py` file into the `skill_env` folder and create a .zip file of the contents of the folder (**not** the folder itself). Name the file `skill.zip`. You can check the AWS Lambda docs to get more information on creating a deployment package.

Before uploading the code to AWS Lambda, you need to create an AWS Lambda function and create the skill on the Alexa Developer Portal.

1.2.8 Creating an AWS Lambda function

Refer to Hosting a Custom Skill as an AWS Lambda Function for a walkthrough on creating an AWS Lambda function with the correct role for your skill. When creating the function, select the *Author from scratch* option and select the `Python 2.7` or `Python 3.6` runtime.

Once you've created your AWS Lambda function, it's time to give the Alexa service the ability to invoke it. To do this, navigate to the **Triggers** tabs in your Lambda's configuration, and add **Alexa Skills Kit** as the trigger type. Once this is done, upload the `skill.zip` file produced in the previous step and fill in the *handler* information with module_name.handler which is `hello_world.handler` for this example.

1.2.9 Configuring and testing your skill

Now that the skill code has been uploaded to AWS Lambda, you can configure the skill with Alexa.

- Create a new skill by following these steps:

  1. Log in to the Alexa Skills Kit Developer Console.
  2. Click the **Create Skill** button in the upper right.
  3. Enter "HelloWorld" as your skill name and click Next.
  4. For the model, select **Custom** and click **Create skill**.

- Next, define the interaction model for the skill. Select the **Invocation** option from the sidebar and enter "greeter" for the **Skill Invocation Name**.

- Next, add an intent called `HelloWorldIntent` to the interaction model. Click the **Add** button under the Intents section of the Interaction Model. Leave "**Create custom intent**" selected, enter "**HelloWorldIntent**" for the intent name, and create the intent. On the intent detail page, add some sample utterances that users can say to invoke the intent. For this example, consider the following sample utterances, and feel free to add others.

```
say hello
say hello world
hello
say hi
say hi world
hi
how are you
```

- Since `AMAZON.CancelIntent`, `AMAZON.HelpIntent`, and `AMAZON.StopIntent` are built-in Alexa intents, you do not need to provide sample utterances for them.
- The Developer Console allows you to edit the entire skill model in JSON format. Select **JSON Editor** from the sidebar. For this sample, you can use the following JSON schema.

```
{
  "interactionModel": {
    "languageModel": {
      "invocationName": "greeter",
      "intents": [
        {
          "name": "AMAZON.CancelIntent",
          "samples": []
        },
        {
          "name": "AMAZON.HelpIntent",
          "samples": []
        },
        {
          "name": "AMAZON.StopIntent",
          "samples": []
        },
        {
          "name": "HelloWorldIntent",
          "slots": [],
          "samples": [
            "how are you",
            "hi",
            "say hi world",
            "say hi",
            "hello",
            "say hello world",
            "say hello"
          ]
        }
      ],
      "types": []
    }
  }
}
```

- Once you are done editing the interaction model, be sure to save and build the model.
- Next, configure the endpoint for the skill. To do this, follow these steps:

1. Under your skill, click the **Endpoint** tab, select AWS Lambda ARN, and copy the **Skill ID** of the skill you just created.
2. Open the AWS Developer Console in a new tab.
3. Navigate to the AWS Lambda function created in the previous step.
4. From the **Designer** menu, add the **Alexa Skills Kit** trigger menu, and scroll down to paste the skill ID into the **Skill ID Verification** configuration. Click **Add and save** once completed to update the AWS Lambda function.
5. Copy the AWS Lambda function **ARN** from the top right corner of the page. An ARN is a unique resource number that helps Alexa service identify the AWS Lambda function it needs to call during skill invocation.
6. Navigate to the Alexa Skills Kit Developer Console, and click on your **HelloWorld** skill.
7. Under your skill, click **Endpoint** tab, select **AWS Lambda ARN** and paste in the ARN under **Default Region** field.
8. The rest of the settings can be left at their default values. Click **Save Endpoints**.
9. Click **Invocation** tab, save and build the model.

- At this point you can test the skill. In the top navigation, click **Test**. Make sure that the **Test is enabled for this skill** option is enabled. You can use the Test page to simulate requests, in text and voice form.
- Use the invocation name along with one of the sample utterances as a guide. For example, *tell greeter to say hello* should result in your skill responding with “Hello World” voice and “Hello World” card on devices with display. You can also open the Alexa app on your phone or at https://alexa.amazon.com) and see your skill listed under **Your Skills**.
- Feel free to start experimenting with your intents as well as the corresponding request handlers in your skill’s code. Once you’re finished iterating, optionally move on to getting your skill certified and published so it can be used by customers worldwide.

1.3 Sample Skills

This section provides sample skills that demonstrate the usage of ASK SDK for Python to build engaging Alexa Skills.

1.3.1 Hello World (using Classes)

This code sample will allow you to hear a response from Alexa when you trigger it. It is a minimal sample to get you familiarized with the Alexa Skills Kit and AWS Lambda. This sample shows how to create a skill using the Request Handler classes. For more information, check the Request Processing documentation.

1.3.2 Hello World (using Decorators)

This code sample will allow you to hear a response from Alexa when you trigger it. It is a minimal sample to get you familiarized with the Alexa Skills Kit and AWS Lambda. This sample shows how to create a skill using the Request Handler Decorators. For more information, check the Request Processing documentation.

1.3.3 Color Picker

This is a step-up in functionality from Hello World. When the user provides their favorite color, Alexa remembers it and tells the user their favorite color. It allows you to capture input from your user and demonstrates the use of Slots. It also demonstrates use of session attributes and request, response interceptors.

1.3.4 Fact

Template for a basic fact skill. You'll provide a list of interesting facts about a topic, Alexa will select a fact at random and tell it to the user when the skill is invoked. Demonstrates use of multiple locales and internationalization in the skill.

1.3.5 Quiz Game

Template for a basic quiz game skill. Alexa quizzes the user with facts from a list you provide. Demonstrates use of render template directives to support displays on Alexa-enabled devices with a screen.

1.3.6 Device Address

Sample skill that shows how to request and access the configured address in the user's device settings. Demonstrates how to use the alexa APIs using the SDK. For more information, check the documentation on Alexa Service Clients

1.3.7 Fact with In-Skill Purchases

Sample fact skill with in-skill purchase features, by offering different packs of facts behind a purchase, and a subscription to unlock all of the packs at once. Demonstrates calling monetization alexa service and using ASK CLI to enable in-skill purchasing.

1.3.8 City Guide

Template for a local recommendations skill. Alexa uses the data that you provide to offer recommendations according to the user's stated preferences. Demonstrates calling external APIs from the skill.

1.3.9 Pet Match

Sample skill that matches the user with a pet. Alexa prompts the user for the information it needs to determine a match. Once all of the required information is collected, the skill sends the data to an external web service that processes the data and returns the match. Demonstrates how to prompt and parse multiple values from customers using Dialog Management and Entity Resolution.

1.3.10 High Low Game

Template for a basic high-low game skill. When the user guesses a number, Alexa tells the user whether the number she has in mind is higher or lower. Demonstrates use of persistence attributes and the persistence adapter in the SDK.

1.3.11 AudioPlayer SingleStream and MultiStream

Sample skills that show how to use AudioPlayer interface and PlaybackController interface in Alexa, to build audio-player skills. The SingleStream skill sample demonstrates how to create a live radio skill, along with localization support. The MultiStream skill sample demonstrates how to create a basic podcast skill that can play multiple, pre-recorded audio streams.

1.4 Alexa Capabilities supported by SDK

This section provides all the Alexa Capabilities that are currently supported in the SDK.

1.4.1 Stable

- Amazon Pay
- Audio Player
- Display – Body templates for devices with a screen
- GadgetsGame Engine – Echo Buttons
- Directive Service (Progressive Response)
- Messaging
- Monetization
- Video
- Device Address
- Lists
- Request for customer contact information
- Obtain customer settings information
- Account Linking
- Entity Resolution
- Dialog Management
- Location Services
- Reminders

1.4.2 Preview

Note: The following capabilities are released as Preview. The interfaces might change during a stable release.

- Connections
- Alexa Presentation Language
- Name-free Interactions

1.5 Setting Up The ASK SDK

This guide will show you how to include the SDK as a dependency in your Python project.

1.6 Developing Your First Skill

Walks you through step-by-step instructions for building the Hello World sample.

1.7 Sample Skills

Provides a comprehensive list of skill samples using the SDK.

1.8 SDK Supported Alexa Capabilities

Provides a list of alexa capabilities, that are supported in the ASK SDK.

CHAPTER 2

SDK Features

2.1 Request Processing

2.1.1 Standard Request

Alexa communicates with the skill service via a request-response mechanism using HTTP over SSL/TLS. When a user interacts with an Alexa skill, your service receives a POST request containing a JSON body. The request body contains the parameters necessary for the service to perform its logic and generate a JSON-formatted response. The documentation on JSON structure of the request body can be found here.

Though Python can handle JSON natively as `dict` objects, for providing type support, they are deserialized into model objects (`ask-sdk-model` package) for skill consumption.

2.1.2 Handler Input

Request Handlers, Request and Response Interceptors, and Exception Handlers are all passed a global `HandlerInput` object during invocation. This object exposes various entities useful in request processing, including:

- **request_envelope**: Contains the entire request body sent to skill.
- **attributes_manager**: Provides access to request, session, and persistent attributes.
- **service_client_factory**: Constructs service clients capable of calling Alexa APIs.
- **response_builder**: Contains helper function to build responses.
- **context**: Provides an optional, context object passed in by the host container. For example, for skills running on AWS Lambda, this is the context object for the AWS Lambda function.

2.1.3 Request Handlers

Request handlers are responsible for handling one or more types of incoming alexa requests. There are two ways of creating custom request handlers:

- By implementing the `AbstractRequestHandler` class.
- By decorating a custom handle function using the Skill Builder `request_handler` decorator.

> **Warning:** You may use either implementation using **classes** or **decorators** to write a skill.
>
> We strongly recommend you to choose **one** of the options and use it consistently throughout your skill, for better code structure.

Interface

AbstractRequestHandler Class

If you plan on using the `AbstractRequestHandler` class, you will need to implement the following methods :

- **can_handle**: `can_handle` method is called by the SDK to determine if the given handler is capable of processing the incoming request. This function accepts a *Handler Input* object and expects a boolean to be returned. If the method returns **True**, then the handler is supposed to handle the request successfully. If it returns **False**, the handler is not supposed to handle the input request and hence not executed to completion. Because of the various attributes in `HandlerInput` object, you can write any condition to let SDK know whether the request can be handled gracefully or not.
- **handle**: `handle` method is called by the SDK when invoking the request handler. This function contains the handler's request processing logic, accepts *Handler Input* and returns a `Response` object.

```
class AbstractRequestHandler(object):
    @abstractmethod
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        pass

    @abstractmethod
    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        pass
```

RequestHandler Decorator

The `request_handler` decorator from SkillBuilder class is a custom wrapper on top of the `AbstractRequestHandler` class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator expects a `can_handle_func` parameter. This is similar to the `can_handle` method in `AbstractRequestHandler`. The value passed should be a function that accepts a *Handler Input* object and returns a `boolean` value
- The decorated function should accept only one parameter, which is the *Handler Input* object and may return a `Response` object.

```
class SkillBuilder(object):
    ....
    def request_handler(self, can_handle_func):
```

(continues on next page)

(continued from previous page)

```
        def wrapper(handle_func):
            # wrap the can_handle and handle into a class
            # add the class into request handlers list
            ....
        return wrapper
```

Code Sample

The following example shows a request handler class that can handle the HelloWorldIntent.

AbstractRequestHandler Class

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.ui import SimpleCard

class HelloWorldIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        return is_intent_name("HelloWorldIntent")(handler_input)

    def handle(self, handler_input):
        speech_text = "Hello World";

        return handler_input.response_builder.speak(speech_text).set_card(
            SimpleCard("Hello World", speech_text)).response
```

The can_handle function detects if the incoming request is an IntentRequest and returns true if the intent name is HelloWorldIntent. The handle function generates and returns a basic "Hello World" response.

RequestHandler Decorator

```
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.ui import SimpleCard
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

@sb.request_handler(can_handle_func = is_intent_name("HelloWorldIntent"))
def hello_world_intent_handler(handler_input):
    speech_text = "Hello World!"

    return handler_input.response_builder.speak(speech_text).set_card(
        SimpleCard("Hello World", speech_text)).response
```

The is_intent_name function accepts a string parameter and returns an anonymous function which accepts a HandlerInput as input parameter and checks if the incoming request in HandlerInput is an IntentRequest and returns if the intent name is the passed in string, which is HelloWorldIntent in this example. The handle function generates and returns a basic "Hello World" response.

Registering and Processing the Request Handlers

The SDK calls the can_handle function on its request handlers in the order in which they were provided to the Skill builder.

AbstractRequestHandler Class

If you are following the `AbstractRequestHandler` class approach, then you can register the request handlers in the following way

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# Implement FooHandler, BarHandler, BazHandler classes

sb.add_request_handler(FooHandler())
sb.add_request_handler(BarHandler())
sb.add_request_handler(BazHandler())
```

RequestHandler Decorator

If you are following the `request_handler` decorator approach, then there is no need to explicitly register the handler functions, since they are already decorated using a skill builder instance.

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# decorate foo_handler, bar_handler, baz_handler functions
```

Note: In the above example, the SDK calls request handlers in the following order:

1. `FooHandler` class / `foo_handler` function
2. `BarHandler` class / `bar_handler` function
3. `BazHandler` class / `baz_handler` function

The SDK always chooses the first handler that is capable of handling a given request. In this example, if both `FooHandler` class /`foo_handler` function and `BarHandler` class /`bar_handler` function are capable of handling a particular request, `FooHandler` class /`foo_handler` function is always invoked. Keep this in mind when designing and registering request handlers.

2.1.4 Exception Handlers

Exception handlers are similar to request handlers, but are instead responsible for handling one or more types of exceptions. They are invoked by the SDK when an unhandled exception is thrown during the course of request processing.

In addition to the *Handler Input* object, the handler also has access to the exception raised during handling the input request, thus making it easier for the handler to figure out how to handle the corresponding exception.

Similar to *Request Handlers*, custom request interceptors can be implemented in two ways:

- By implementing the `AbstractExceptionHandler` class.
- By decorating a custom exception handling function using the Skill Builder `exception_handler` decorator.

> **Warning:** You may use either implementation using **classes** or **decorators** to write a skill.
>
> We strongly recommend you to choose **one** of the options and use it consistently throughout your skill, for better code structure.

Interface

AbstractExceptionHandler Class

If you plan on using the `AbstractExceptionHandler` class, you will need to implement the following methods :

- **can_handle**: `can_handle` method, which is called by the SDK to determine if the given handler is capable of handling the exception. This function returns **True** if the handler can handle the exception, or **False** if not. Return `True` in all cases to create a catch-all handler.
- **handle**: `handle` method, which is called by the SDK when invoking the exception handler. This function contains all exception handling logic, and returns a `Response` object.

```
class AbstractExceptionHandler(object):
    @abstractmethod
    def can_handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> bool
        pass

    @abstractmethod
    def handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> Response
        pass
```

ExceptionHandler Decorator

The `exception_handler` decorator from SkillBuilder class is a custom wrapper on top of the `AbstractExceptionHandler` class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator expects a `can_handle_func` parameter. This is similar to the `can_handle` method in `AbstractExceptionHandler`. The value passed should be a function that accepts a *Handler Input* object, an `Exception` instance and returns a `boolean` value.
- The decorated function should accept only two parameters, the *Handler Input* object and `Exception` object. It may return a `Response` object.

```
class SkillBuilder(object):
    ....
    def exception_handler(self, can_handle_func):
        def wrapper(handle_func):
            # wrap the can_handle and handle into a class
            # add the class into exception handlers list
            ....
        return wrapper
```

Code Sample

The following example shows an exception handler that can handle any exception with name that contains “AskSdk”.

AbstractExceptionHandler Class

```
class AskExceptionHandler(AbstractExceptionHandler):
    def can_handle(self, handler_input, exception):
        return 'AskSdk' in exception.__class__.__name__

    def handle(self, handler_input, exception):
        speech_text = "Sorry, I am unable to figure out what to do. Try again later!!"

        return handler_input.response_builder.speak(speech_text).response
```

The handler's `can_handle` method returns True if the incoming exception has a name that starts with "AskSdk". The `handle` method returns a graceful exception response to the user.

ExceptionHandler Decorator

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

@sb.exception_handler(can_handle_func = lambda i, e: 'AskSdk' in e.__class__.__name__)
def ask_exception_intent_handler(handler_input, exception):
    speech_text = "Sorry, I am unable to figure out what to do. Try again later!!"

    return handler_input.response_builder.speak(speech_text).response
```

Registering and Processing the Exception Handlers

AbstractExceptionHandler Class

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# Implement FooExceptionHandler, BarExceptionHandler, BazExceptionHandler classes

sb.add_exception_handler(FooExceptionHandler())
sb.add_exception_handler(BarExceptionHandler())
sb.add_exception_handler(BazExceptionHandler())
```

ExceptionHandler Decorator

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# decorate foo_exception_handler, bar_exception_handler, baz_exception_handler␣
↪functions
```

Note: Like request handlers, exception handlers are executed in the order in which they were registered to the Skill.

2.1.5 Request and Response Interceptors

The SDK supports Global Request and Response Interceptors that execute **before** and **after** matching `RequestHandler` execution, respectively.

Request Interceptors

The Global Request Interceptor accepts a Handler Input object and processes it, before processing any of the registered request handlers. Similar to *Request Handlers*, custom request interceptors can be implemented in two ways:

- By implementing the `AbstractRequestInterceptor` class.
- By decorating a custom process function using the Skill Builder `global_request_interceptor` decorator.

Warning: You may use either implementation using **classes** or **decorators** to write a skill.

We strongly recommend you to choose **one** of the options and use it consistently throughout your skill, for better code structure.

Interface

AbstractRequestInterceptor Class

The `AbstractRequestInterceptor` class usage needs you to implement the `process` method. This method takes a *Handler Input* instance and doesn't return anything.

```
class AbstractRequestInterceptor(object):
    @abstractmethod
    def process(self, handler_input):
        # type: (HandlerInput) -> None
        pass
```

GlobalRequestInterceptor Decorator

The `global_request_interceptor` decorator from SkillBuilder class is a custom wrapper on top of the `AbstractRequestInterceptor` class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator should be invoked as a function rather than as a function name, since it requires the skill builder instance, to register the interceptor.
- The decorated function should accept only one parameter, which is the *Handler Input* object and the return value from the function is not captured.

```
class SkillBuilder(object):
    ....
    def global_request_interceptor(self):
        def wrapper(process_func):
            # wrap the process_func into a class
            # add the class into request interceptors list
            ....
        return wrapper
```

Code Sample

The following example shows a request interceptor class that can print the request received by Alexa service, in AWS CloudWatch logs, before handling it.

AbstractRequestInterceptor Class

```
from ask_sdk_core.dispatch_components import AbstractRequestInterceptor

class LoggingRequestInterceptor(AbstractRequestInterceptor):
    def process(self, handler_input):
        print("Request received: {}".format(handler_input.request_envelope.request))
```

GlobalRequestInterceptor Decorator

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

@sb.global_request_interceptor()
def request_logger(handler_input):
    print("Request received: {}".format(handler_input.request_envelope.request))
```

Registering and Processing the Request Interceptors

Request interceptors are invoked immediately before execution of the request handler for an incoming request. Request attributes in *Handler Input*'s `Attribute Manager` provide a way for request interceptors to pass data and entities on to other request interceptors and request handlers.

AbstractRequestInterceptor Class

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# Implement FooInterceptor, BarInterceptor, BazInterceptor classes

sb.add_global_request_interceptor(FooInterceptor())
sb.add_global_request_interceptor(BarInterceptor())
sb.add_global_request_interceptor(BazInterceptor())
```

GlobalRequestInterceptor Decorator

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()
# decorate foo_interceptor, bar_interceptor, baz_interceptor functions
```

Note: In the above example, the SDK executes all request interceptors in the following order:

1. `FooInterceptor` class / `foo_interceptor` function
2. `BarInterceptor` class / `bar_interceptor` function
3. `BazInterceptor` class / `baz_interceptor` function

Response Interceptors

The Global Response Interceptor accepts a *Handler Input* object, a *Response* and processes them, after executing the supported request handler. Similar to *Request Interceptors*, custom response interceptors can be implemented in two ways:

- By implementing the AbstractResponseInterceptor class.
- By decorating a custom process function using the Skill Builder global_response_interceptor decorator.

Warning: You may use either implementation using **classes** or **decorators** to write a skill.

We strongly recommend you to choose **one** of the options and use it consistently throughout your skill, for better code structure.

Interface

AbstractResponseInterceptor Class

The AbstractResponseInterceptor class usage needs you to implement the process method. This method takes a *Handler Input* instance, a Response object that is returned from the previously executed request handler. The method doesn't return anything.

```
class AbstractResponseInterceptor(object):
    @abstractmethod
    def process(self, handler_input, response):
        # type: (HandlerInput, Response) -> None
        pass
```

GlobalResponseInterceptor Decorator

The global_response_interceptor decorator from SkillBuilder class is a custom wrapper on top of the AbstractResponseInterceptor class and provides the same functionality to any custom decorated function. However, there are couple of things to take into consideration, before using the decorator:

- The decorator should be invoked as a function rather than as a function name, since it requires the skill builder instance, to register the interceptor.
- The decorated function should accept two parameters, which are the *Handler Input* object and Response object respectively. The return value from the function is not captured.

```
class SkillBuilder(object):
    ....
    def global_response_interceptor(self):
        def wrapper(process_func):
            # wrap the process_func into a class
            # add the class into response interceptors list
            ....
        return wrapper
```

Code Sample

The following example shows a response interceptor class that can print the response received from successfully handling the request, in AWS CloudWatch logs, before returning it to the Alexa Service.

AbstractRequestInterceptor Class

```
from ask_sdk_core.dispatch_components import AbstractResponseInterceptor

class LoggingResponseInterceptor(AbstractResponseInterceptor):
```

(continues on next page)

(continued from previous page)

```
    def process(handler_input, response):
        print("Response generated: {}".format(response))
```

GlobalRequestInterceptor Decorator

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

@sb.global_response_interceptor()
def response_logger(handler_input, response):
    print("Response generated: {}".format(response))
```

Registering and Processing the Response Interceptors

Response interceptors are invoked immediately after execution of the request handler for an incoming request.

AbstractRequestInterceptor Class

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# Implement FooInterceptor, BarInterceptor, BazInterceptor classes

sb.add_global_response_interceptor(FooInterceptor())
sb.add_global_response_interceptor(BarInterceptor())
sb.add_global_response_interceptor(BazInterceptor())
```

GlobalRequestInterceptor Decorator

```
from ask_sdk_core.skill_builder import SkillBuilder

sb = SkillBuilder()

# decorate foo_interceptor, bar_interceptor, baz_interceptor functions
```

Note: Similar to the processing of *Request Interceptors*, all of the response interceptors are executed in the same order they are registered.

2.2 Response Building

2.2.1 Standard Response

If you are using the AWS lambda as your skill endpoint, you are only responsible for providing the response body in order for Alexa to respond to a customer request. The documentation on the JSON structure of the response body can be found here.

A response body may contain the following properties:

- version

- sessionAttributes
- response

ASK SDK for Python helps filling the version and sessionAttributes so you can focus on building the response instead of writing boilerplate code.

Similar to standard requests, the SDK creates response objects as deserialized model objects (`ask-sdk-model` package) and internally handles serializing them to response JSON before sending to Alexa Service.

2.2.2 Response Factory

The SDK includes a `ResponseFactory` class, that contains helper functions for constructing responses. A `Response` may contain multiple elements, and the helper functions aid in generating responses, reducing the need to initialize and set the elements of each response.

Interface

```
class ResponseFactory(object):
    def __init__(self):
        self.response = ....  # Response object

    def speak(self, speech):
        # type: (str) -> 'ResponseFactory'
        ....

    def ask(self, speech):
        # type: (str) -> 'ResponseFactory'
        ....

    def set_card(self, card):
        # type: (ask_sdk_model.ui.card.Card) -> 'ResponseFactory'
        ....

    def add_directive(self, directive):
        # type: (ask_sdk_model.directive.Directive) -> 'ResponseFactory'
        ....

    def set_should_end_session(self, end_session):
        # type: (bool) -> 'ResponseFactory'
        ....
```

`response_builder`, an instance of the `ResponseFactory` class, is provided to the skill developers through the HandlerInput object, which is the standard argument passed to the skill components.

Tip: The contents of the speak and reprompt values get wrapped in SSML tags. Build engaging conversations using supported SSML tags directly in your speech text.

Note:

- For using and adding different directives, look at the Directive model definition.
- For using and setting a card, look at the Card model definition.

Sample Code

The following example shows how to construct a response containing a `StandardCard` and a `BodyTemplate2` display object through `handler_input.response_builder`.

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_core.response_helper import get_plain_text_content

from ask_sdk_model.response import Response
from ask_sdk_model.interfaces.display import (
    ImageInstance, Image, RenderTemplateDirective,
    BackButtonBehavior, BodyTemplate2)
from ask_sdk_model import ui

class HelloIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("HelloIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        response_builder = handler_input.response_builder

        speech = "This is a sample response"

        response_builder.set_card(
            ui.StandardCard(
                title="Card Title",
                text="Hey this is a sample card",
                image=ui.Image(
                    small_image_url="<Small Image URL>",
                    large_image_url="<Large Image URL>"
                )
            )
        )

        if supports_display(handler_input):
            img = Image(
                sources=[ImageInstance(url="<Large Image URL>")])
            title = "Template Title"
            primary_text = get_plain_text_content(
                primary_text="some text")

            response_builder.add_directive(
                RenderTemplateDirective(
                    BodyTemplate2(
                        back_button=BackButtonBehavior.VISIBLE,
                        image=img, title=title,
                        text_content=primary_text)))

        return response_builder.speak(speech).response
```

Text Helpers

The following helper functions are provided to skill developers, to help with text content generation:

get_plain_text_content

```
def get_plain_text_content(primary_text, secondary_text, tertiary_text):
    # type: (str, str, str) -> TextContent
    # Create a text content object with text as PlainText type
    ....
```

get_rich_text_content

```
def get_rich_text_content(primary_text, secondary_text, tertiary_text):
    # type: (str, str, str) -> TextContent
    # Create a text content object with text as RichText type
    ....
```

get_text_content

```
def get_text_content(
    primary_text, primary_text_type,
    secondary_text, secondary_text_type,
    tertiary_text, tertiary_text_type):
    # type: (str, str, str, str, str, str) -> TextContent
    # Create a text content object with text as corresponding passed-type
    # Passed-in type is defaulted to PlainText
    ....
```

2.3 Skill Attributes

This guide provides information on different scopes of attributes available to the skill developer, and how to use them in the skill.

2.3.1 Attributes

The SDK allows you to store and retrieve attributes at different scopes. For example, attributes can be used to store data that you retrieve on subsequent requests. You can also use attributes in your handler's `can_handle` logic to add conditions during request routing.

An attribute consists of a key and a value. The key is enforced as a `str` type and the value is an unbounded `object`. For session and persistent attributes, you must ensure that value types are serializable so they can be properly stored for subsequent retrieval. This restriction does not apply to request-level attributes because they do not persist outside of the request processing lifecycle.

2.3.2 Attribute Scopes

Request Attributes

Request attributes only last within a single request processing lifecycle. Request attributes are initially empty when a request comes in, and are discarded once a response has been produced.

Request attributes are useful with request and response interceptors. For example, you can inject additional data and helper methods into request attributes through a request interceptor so they are retrievable by request handlers.

Session Attributes

Session attributes persist throughout the lifespan of the current skill session. Session attributes are available for use with any in-session request. Any attributes set during the request processing lifecycle are sent back to the Alexa service and provided in the next request in the same session.

Session attributes do not require the use of an external storage solution. They are not available for use when handling out-of-session requests. They are discarded once the skill session closes.

Note: Since session attributes are stored in the `session` property of Alexa's *`ask_sdk_model.request_envelope.RequestEnvelope`* and *`ask_sdk_model.response_envelope.ResponseEnvelope`* objects, only serializable types can be stored under them. The *`ask_sdk_core.serialize.DefaultSerializer`* is used to serialize / deserialize the values.

Persistent Attributes

Persistent attributes persist beyond the lifecycle of the current session. How these attributes are stored, including key scope (user ID or device ID), TTL, and storage layer depends on the configuration of the skill.

Note: Persistent attributes are only available when you configure the skill instance with a `PersistenceAdapter`. A call to the `AttributesManager` to retrieve or save persistent attributes will raise an exception if the `PersistenceAdapter` has not been configured.

2.3.3 AttributesManager

The `AttributesManager` exposes attributes that you can retrieve and update in your handlers. `AttributesManager` is available to handlers via the Handler Input object. The `AttributesManager` takes care of attributes retrieval and saving so that you can interact directly with attributes needed by your skill.

Interface

```
class AttributesManager(object):
    def __init__(self, request_envelope, persistence_adapter=None):
        # type: (RequestEnvelope, AbstractPersistenceAdapter) -> None
        ....

    @property
    def request_attributes(self):
        # type: () -> Dict[str, Any]
        # Request Attributes getter
        ....

    @request_attributes.setter
    def request_attributes(self, attributes):
        # type: (Dict[str, Any]) -> None
        # Request Attributes setter
```

(continues on next page)

(continued from previous page)

```
        ....

    @property
    def session_attributes(self):
        # type: () -> Dict[str, Any]
        # Session Attributes getter
        ....

    @session_attributes.setter
    def session_attributes(self, attributes):
        # type: (Dict[str, Any]) -> None
        # Session Attributes setter
        ....

    @property
    def persistent_attributes(self):
        # type: () -> Dict[str, Any]
        # Persistence Attributes getter
        # Uses the Persistence adapter to get the attributes
        ....

    @persistent_attributes.setter
    def persistent_attributes(self, attributes):
        # type: (Dict[str, Any]) -> None
        # Persistent Attributes setter
        ....

    def save_persistent_attributes(self):
        # type: () -> None
        # Persistence Attributes save
        # Save the Persistence adapter to save the attributes
        ....
```

The following example shows how you can retrieve and save persistent attributes.

```
class PersistenceAttributesHandler(AbstractRequestHandler):
    def can_handle(handler_input):
        persistence_attr = handler_input.attributes_manager.persistent_attributes
        return persistence_attr['foo'] == 'bar'

    def handle(handler_input):
        persistence_attr = handler_input.attributes_manager.persistent_attributes
        persistence_attr['foo'] = 'baz'
        handler_input.attributes_manager.save_persistent_attributes()
        return handler_input.response_builder.response
```

Note: To improve skill performance, `AttributesManager` caches the persistent attributes locally. `persistent_attributes` setter will only update the locally cached persistent attributes. You need to call `save_persistent_attributes()` to save persistent attributes to the persistence layer.

2.3.4 PersistenceAdapter

The `AbstractPersistenceAdapter` is used by `AttributesManager` when retrieving and saving attributes to persistence layer (i.e. database or local file system). You can register any customized `PersistenceAdapter` that conforms to the AbstractPersistenceAdapter interface with the SDK.

All implementations of AbstractPersistenceAdapter needs to follow the following interface.

Interface

```
class AbstractPersistenceAdapter(object):
    def get_attributes(self, request_envelope):
        # type: (RequestEnvelope) -> Dict[str, Any]
        pass

    def save_attributes(self, request_envelope, attributes):
        # type: (RequestEnvelope, Dict[str, Any]) -> None
        pass
```

DynamoDbPersistenceAdapter

The ask-sdk-dynamodb-persistence-adapter package provides an implementation of AbstractPersistenceAdapter using AWS DynamoDB.

Interface

```
from ask_sdk_dynamodb.adapter import DynamoDbAdapter

adapter = DynamoDbAdapter(table_name, partition_key_name="id",
            attribute_name="attributes", create_table=False,
            partition_keygen=user_id_partition_keygen,
            dynamodb_resource=boto3.resource("dynamodb"))
```

Configuration Options

- **table_name** (string) - The name of the DynamoDB table used.
- **partition_key_name** (string) - Optional. The name of the partition key column. Default to "id" if not provided.
- **attributes_name** (string) - Optional. The name of the attributes column. Default to "attributes" if not provided.
- **create_table** (boolean) - Optional. Set to True to have DynamoDbAdapter automatically create the table if it does not exist. Default to False if not provided.
- **partition_keygen** (callable) - Optional. The function used to generate partition key using RequestEnvelope. Default to generate the partition key using the user_id.
- **dynamodb_resource** (AWS.DynamoDB ServiceResource) - Optional. The DynamoDBClient used to query AWS DynamoDB table. You can inject your DynamoDBClient with custom configuration here. Default to use boto3.resource("dynamodb").

2.4 Alexa Service Clients

Alexa Skills Kit provides multiple service APIs that you can use to personalize your skill experience. The SDK includes service clients that you can use to call Alexa APIs from within your skill logic.

2.4.1 ServiceClientFactory

The `service_client_factory` contained inside the Handler Input allows you to retrieve client instances for every supported Alexa service. It takes care of creating individual service clients and configuring the metadata like `api_access_token` and `api_endpoint`.

Since it is available in `handler_input` through `service_client_factory` attribute, service clients can be used in any request handler, exception handler, and request, response interceptors.

Available service clients

```
def get_device_address_service(self):
    # type: () -> ask_sdk_model.services.device_address.DeviceAddressServiceClient

def get_directive_service(self):
    # type: () -> ask_sdk_model.services.directive.DirectiveServiceClient

def get_list_management_service(self):
    # type: () -> ask_sdk_model.services.list_management.ListManagementServiceClient

def get_monetization_service(self):
    # type: () -> ask_sdk_model.services.monetization.MonetizationServiceClient

def get_ups_service(self):
    # type: () -> ask_sdk_model.services.ups.UpsServiceClient
```

Note: The `service_client_factory` is only available for use, when you configure the skill instance with an `ApiClient`.

2.4.2 ApiClient

The `ask_sdk_model.services.api_client.ApiClient` is used by the `service_client_factory` when making API calls to Alexa services. You can register any customized `ApiClient` that conforms to the following interface with the SDK.

Interface

```
class ask_sdk_model.services.api_client.ApiClient:
    def invoke(self, request):
        # type: (ApiClientRequest) -> ApiClientResponse

class ask_sdk_model.services.api_client_request.ApiClientRequest(ApiClientMessage):
    def __init__(self, headers=None, body=None, url=None, method=None):
        # type: (List[Tuple[str, str]], str, str, str) -> None
```

(continues on next page)

(continued from previous page)

```
class ask_sdk_model.services.api_client_request.ApiClientResponse(ApiClientMessage):
    def __init__(self, headers=None, body=None, status_code=None):
        # type: (List[Tuple[str, str]], str, int) -> None

class ask_sdk_model.services.api_client_message.ApiClientMessage(object):
    def __init__(self, headers=None, body=None):
        # type: (List[Tuple[str, str]], str) -> None
```

The CustomSkillBuilder constructor can be used to register the ApiClient.

```
from ask_sdk_core.skill_builder import CustomSkillBuilder

sb = CustomSkillBuilder(api_client=<YourClassInstance>)
```

DefaultApiClient

A `DefaultApiClient` based on the `requests` library, is made available in the `ask_sdk_core.api_client` module for skill developers.

This client is registered by default in the StandardSkillBuilder. Alternatively, skill developers can register this client to the `CustomSkillBuilder`.

```
from ask_sdk_core.skill_builder import CustomSkillBuilder
from ask_sdk_core.api_client import DefaultApiClient

sb = CustomSkillBuilder(api_client=DefaultApiClient())
```

2.4.3 DeviceAddressServiceClient

`DeviceAddressServiceClient` can be used to query Device Address API for address data associated with the customer's Alexa device. You can then use this address data to provide key functionality for the skill, or to enhance the customer experience. For example, your skill could provide a list of nearby store locations or provide restaurant recommendations using this address information.

Interface

```
class ask_sdk_model.services.device_address.DeviceAddressServiceClient:
    def get_country_and_postal_code(device_id):
        # type: (str) -> Union[ShortAddress, Error]

    def get_full_address(self, device_id):
        # type: (str) -> Union[Address, Error]

class ask_sdk_model.services.device_address.ShortAddress:
    def __init__(self, country_code=None, postal_code=None):
        # type: (Optional[str], Optional[str]) -> None

class ask_sdk_model.services.device_address.Address:
    def __init__(
        self, address_line1=None, address_line2=None, address_line3=None,
        country_code=None, state_or_region=None, city=None,
```

(continues on next page)

(continued from previous page)

```
        district_or_county=None, postal_code=None):
        # type: (Optional[str], Optional[str], Optional[str], Optional[str],␣
↪Optional[str], Optional[str], Optional[str], Optional[str]) -> None
```

Note: The device_id can be retrieved from `handler_input.request_envelope.context.system.device.device_id`.

More information on the models can be found here.

Code Sample

The following example shows how a request handler retrieves customer's full address.

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.response import Response
from ask_sdk_model.ui import AskForPermissionsConsentCard
from ask_sdk_model.services import ServiceException

NOTIFY_MISSING_PERMISSIONS = ("Please enable Location permissions in "
                              "the Amazon Alexa app.")
NO_ADDRESS = ("It looks like you don't have an address set. "
              "You can set your address from the companion app.")
ADDRESS_AVAILABLE = "Here is your full address: {}, {}, {}"
ERROR = "Uh Oh. Looks like something went wrong."
LOCATION_FAILURE = ("There was an error with the Device Address API. "
                    "Please try again.")

permissions = ["read::alexa:device:all:address"]

class GetAddressIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("GetAddressIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        req_envelope = handler_input.request_envelope
        service_client_fact = handler_input.service_client_factory
        response_builder = handler_input.response_builder

        if not (req_envelope.context.system.user.permissions and
                req_envelope.context.system.user.permissions.consent_token):
            response_builder.speak(NOTIFY_MISSING_PERMISSIONS)
            response_builder.set_card(
                AskForPermissionsConsentCard(permissions=permissions))
            return response_builder.response

        try:
            device_id = req_envelope.context.system.device.device_id
            device_addr_client = service_client_fact.get_device_address_service()
            addr = device_addr_client.get_full_address(device_id)
```

(continues on next page)

(continued from previous page)

```
            if addr.address_line1 is None and addr.state_or_region is None:
                response_builder.speak(NO_ADDRESS)
            else:
                response_builder.speak(ADDRESS_AVAILABLE.format(
                    addr.address_line1, addr.state_or_region, addr.postal_code))
            return response_builder.response
        except ServiceException:
            response_builder.speak(ERROR)
            return response_builder.response
        except Exception as e:
            raise e
```

2.4.4 DirectiveServiceClient

`DirectiveServiceClient` can be used to send directives to Progressive Response API. Progressive responses can be used to keep the user engaged while your skill prepares a full response to the user's request.

Interface

```
class ask_sdk_model.services.directive.DirectiveServiceClient:
    def enqueue(self, send_directive_request):
        # type: (SendDirectiveRequest) -> Union[Error]

class ask_sdk_model.services.directive.SendDirectiveRequest:
    def __init__(self, header=None, directive=None):
        # type: (Optional[Header], Optional[SpeakDirective]) -> None

class ask_sdk_model.services.directive.SpeakDirective:
    def __init__(self, speech=None):
        # type: (Optional[str]) -> None
```

More information on the models can be found here.

Code Sample

The following example shows a function that can be used in a `handle` method for sending a progressive response.

```
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_model.services.directive import (
    SendDirectiveRequest, Header, SpeakDirective)
import time

def get_progressive_response(handler_input):
    # type: (HandlerInput) -> None
    request_id_holder = handler_input.request_envelope.request.request_id
    directive_header = Header(request_id=request_id_holder)
    speech = SpeakDirective(speech="Ok, give me a minute")
    directive_request = SendDirectiveRequest(
        header=directive_header, directive=speech)

    directive_service_client = handler_input.service_client_factory.get_directive_
↪service()
```

(continues on next page)

(continued from previous page)

```
    directive_service_client.enqueue(directive_request)
    time.sleep(5)
    return
```

2.4.5 ListManagementServiceClient

ListManagementServiceClient can be used to access the List Management API n order to read or modify both the Alexa default lists and any custom lists customer may have.

Interface

```
class ask_sdk_model.services.list_management.ListManagementServiceClient:
    def get_lists_metadata(self):
        # type: () -> Union[ForbiddenError, Error, AlexaListsMetadata]

    def get_list(self, list_id, status):
        # type: (str, str) -> Union[AlexaList, Error]

    def get_list_item(self, list_id, item_id):
        # type: (str, str) -> Union[AlexaListItem, Error]

    def create_list(self, create_list_request):
        # type: (CreateListRequest) -> Union[Error, AlexaListMetadata]

    def create_list_item(self, list_id, create_list_item_request):
        # type: (str, CreateListItemRequest) -> Union[AlexaListItem, Error]

    def update_list(self, list_id, update_list_request):
        # type: (str, UpdateListRequest) -> Union[Error, AlexaListMetadata]

    def update_list_item(self, list_id, item_id, update_list_item_request):
        # type: (str, str, UpdateListItemRequest) -> Union[AlexaListItem, Error]

    def delete_list(self, list_id):
        # type: (str) -> Union[Error]

    def delete_list_item(self, list_id, item_id):
        # type: (str, str) -> Union[Error]
```

More information on the models can be found here.

2.4.6 MonetizationServiceClient

In-Skill Purchase Service

The SDK provides a MonetizationServiceClient that invokes inSkillPurchase API to retrieve all in-skill products associated with the current skill along with indications if each product is purchasable and/or already purchased by the current customer.

Interface

```
class ask_sdk_model.services.monetization.MonetizationServiceClient:
    def get_in_skill_products(
        self, accept_language, purchasable=None, entitled=None,
        product_type=None, next_token=None, max_results=None):
        # type: (str, Optional[PurchasableState], Optional[EntitledState],␣
↪Optional[ProductType], Optional[str], Optional[float]) -> Union[Error,␣
↪InSkillProductsResponse]

    def get_in_skill_product(self, accept_language, product_id):
        # type: (str, str) -> Union[Error, InSkillProduct]

class ask_sdk_model.services.monetization.InSkillProductsResponse:
    def __init__(self, in_skill_products=None, is_truncated=None, next_token=None):
        # type: (Optional[List[InSkillProduct]], Optional[bool], Optional[str]) ->␣
↪None

class ask_sdk_model.services.monetization.InSkillProduct:
self, product_id=None, reference_name=None, name=None, object_type=None, summary=None,
↪ purchasable=None, entitled=None, active_entitlement_count=None, purchase_mode=None
    def __init__(
        self, product_id=None, reference_name=None, name=None,
        object_type=None, summary=None, purchasable=None, entitled=None,
        active_entitlement_count=None, purchase_mode=None):
        # type: (Optional[str], Optional[str], Optional[str], Optional[ProductType],␣
↪Optional[str], Optional[PurchasableState], Optional[EntitledState], Optional[int],␣
↪Optional[PurchaseMode]) -> None

class ask_sdk_model.services.monetization.ProductType(Enum):
    SUBSCRIPTION = "SUBSCRIPTION"
    ENTITLEMENT = "ENTITLEMENT"
    CONSUMABLE = "CONSUMABLE"

class ask_sdk_model.services.monetization.PurchasableState(Enum):
    PURCHASABLE = "PURCHASABLE"
    NOT_PURCHASABLE = "NOT_PURCHASABLE"

class ask_sdk_model.services.monetization.EntitledState(Enum):
    ENTITLED = "ENTITLED"
    NOT_ENTITLED = "NOT_ENTITLED"

class ask_sdk_model.services.monetization.PurchaseMode(Enum):
    TEST = "TEST"
    LIVE = "LIVE"
```

Note: `accept_language` is the locale of the request and can be retrieved from `handler_input.request_envelope.request.locale`.

More information on the models can be found here.

Code Sample

get_in_skill_products

The `get_in_skill_products` method retrieves all associated in-skill products for the current skill along with purchasability and entitlement indications for each in-skill product for the current skill and customer.

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_request_type
from ask_sdk_model.response import Response
from ask_sdk_model.services.monetization import (
    EntitledState, PurchasableState, InSkillProductsResponse)

class LaunchRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        return is_request_type("LaunchRequest")(handler_input)

    def handle(self, handler_input):
        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()
        product_response = ms.get_in_skill_products(locale)

        if isinstance(product_response, InSkillProductsResponse):
            total_products = len(product_response.in_skill_products)
            entitled_products = len([l for l in product_response.in_skill_products
                                 if l.entitled == EntitledState.ENTITLED])
            purchasable_products = len([l for l in product_response.in_skill_products
                                    if l.purchasable == PurchasableState.PURCHASABLE])

            speech = (
                "Found total {} products of which {} are purchasable and {} "
                "are entitled".format(
                    total_products, purchasable_products, entitled_products))
        else:
            speech = "Something went wrong in loading your purchase history."

        return handler_input.response_builder.speak(speech).response
```

The API response contains an array of in-skill product records.

get_in_skill_product:

The `get_in_skill_product` API retrieves the product record for a single in-skill product identified by a given productId.

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_request_type
from ask_sdk_model.response import Response
from ask_sdk_model.services.monetization import InSkillProduct

class LaunchRequestHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        return is_request_type("LaunchRequest")(handler_input)
```

(continues on next page)

(continued from previous page)

```
    def handle(self, handler_input):
        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()
        product_id = "amzn1.adg.product.<GUID>"
        product_response = ms.get_in_skill_product(locale)

        if isinstance(product_response, InSkillProduct):
            # code to handle InSkillProduct goes here
            speech = ""
            pass
        else:
            speech = "Something went wrong in loading your product."

        return handler_input.response_builder.speak(speech).response
```

The API response contains a single in-skill product record.

More information on these APIs and their usage for skill implementation is available here: Add In-Skill Purchases to a Custom Skill.

In-Skill Purchase Interface

The SDK provides the `set_directive()` method for skills to initiate in-skill purchase and cancellation requests through Alexa. Amazon systems then manage the voice interaction with customers, handle the purchase transaction and return a status response back to the requesting skill. Three different `actions` are supported using this interface:

- `Upsell`
- `Buy`
- `Cancel`

More details about these `actions` and recommended use-cases is available here: Add In-Skill Purchases to a Custom Skill.

Code Sample

Upsell

Skills should initiate the Upsell action to present an in-skill contextually when the user did not explicitly ask for it. E.g. During or after the free content has been served. A productId and upsell message is required to initiate the Upsell action. The upsell message allows developers to specify how Alexa can present the in-skill product to the user before presenting the pricing offer.

```
from ask_sdk_model.interfaces.connections import SendRequestDirective

# In the skill flow, once a decision is made to offer an in-skill product to a
# customer without an explicit ask from the customer


return handler_input.response_builder.add_directive(
        SendRequestDirective(
            name="Upsell",
            payload={
```

(continues on next page)

(continued from previous page)

```
                "InSkillProduct": {
                    "productId": "<product_id>",
                },
                "upsellMessage": "<introductory upsell description for the in-skill␣
↪product>",
            },
            token="correlationToken")
    ).response
```

Buy

Skills should initiate the Buy action when a customer asks to buy a specific in-skill product. A product_id is required to initiate the Buy action.

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.response import Response
from ask_sdk_model.interfaces.connections import SendRequestDirective

# Skills would implement a custom intent (BuyProductIntent below) that captures
# user's intent to buy an in-skill product and then trigger the Buy request for it.
# For e.g. 'Alexa, buy <product name>'

class BuyProductIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("BuyProductIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

        # Obtain the corresponding product_id for the requested in-skill
        # product by invoking InSkillProducts API.
        # The slot variable product_name used below is only for demonstration.

        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()

        product_response = ms.get_in_skill_products(locale)
        slots = handler_input.request_envelope.request.intent.slots
        product_ref_name = slots.get("product_name").value
        product_record = [l for l in product_response.in_skill_products
                          if l.reference_name == product_ref_name]

        if product_record:
            return handler_input.response_builder.add_directive(
                SendRequestDirective(
                    name="Buy",
                    payload={
                        "InSkillProduct": {
                            "productId": product_record[0].product_id
                        }
                    },
                    token="correlationToken")
```

(continues on next page)

(continued from previous page)

```
            ).response
        else:
            return handler_input.response_builder.speak(
                "I am sorry. That product is not available for purchase"
                ).response
```

Cancel

Skills should initiate the Cancel action when a customer asks to cancel an existing Entitlement or Subscription for a supported in-skill product. A product_id is required to initiate the Cancel action.

```
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_core.utils import is_intent_name
from ask_sdk_model.response import Response
from ask_sdk_model.interfaces.connections import SendRequestDirective

# Skills would implement a custom intent (CancelProductIntent below) that captures
# user's intent to cancel an in-skill product and then trigger the Cancel request for␣
↪it.
# For e.g. 'Alexa, cancel <product name>'

class CancelProductIntentHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return is_intent_name("CancelProductIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

        # Obtain the corresponding product_id for the requested in-skill
        # product by invoking InSkillProducts API.
        # The slot variable product_name used below is only for demonstration.

        locale = handler_input.request_envelope.request.locale
        ms = handler_input.service_client_factory.get_monetization_service()

        product_response = ms.get_in_skill_products(locale)
        slots = handler_input.request_envelope.request.intent.slots
        product_ref_name = slots.get("product_name").value
        product_record = [l for l in product_response.in_skill_products
                          if l.reference_name == product_ref_name]

        if product_record:
            return handler_input.response_builder.add_directive(
                SendRequestDirective(
                    name="Cancel",
                    payload={
                        "InSkillProduct": {
                            "productId": product_record[0].product_id
                        }
                    },
                    token="correlationToken")
            ).response
        else:
```

(continues on next page)

(continued from previous page)

```
            return handler_input.response_builder.speak(
                "I am sorry. I don't know that one").response
```

2.4.7 UpsServiceClient

`UpsServiceClient` can be used to query Alexa Customer Profile API for customer contact information and Alexa Customer Settings API for retrieving customer preferences for the time zone, distance measuring unit and temperature measurement unit.

Interface

```
class ask_sdk_model.services.ups.UpsServiceClient:
    def get_profile_email(self):
        # type: () -> Union[str, Error]

    def get_profile_given_name(self):
        # type: () -> Union[str, Error]

    def get_profile_mobile_number(self):
        # type: () -> Union[PhoneNumber, Error]

    def get_profile_name(self):
        # type: () -> Union[str, Error]

    def get_system_distance_units(self, device_id):
        # type: (str) -> Union[Error, DistanceUnits]

    def get_system_temperature_unit(self, device_id):
        # type: (str) -> Union[TemperatureUnit, Error]

    def get_system_time_zone(self, device_id):
        # type: (str) -> Union[str, Error]

class ask_sdk_model.services.ups.PhoneNumber:
    def __init__(self, country_code=None, phone_number=None):
        # type: (Optional[str], Optional[str]) -> None

class ask_sdk_model.services.DistanceUnits(Enum):
    METRIC = "METRIC"
    IMPERIAL = "IMPERIAL"

class ask_sdk_model.services.TemparatureUnit(Enum):
    CELSIUS = "CELSIUS"
    FAHRENHEIT = "FAHRENHEIT"
```

Note: The device_id can be retrieved from `handler_input.request_envelope.context.system.device.device_id`.

More information on the models can be found here.

2.5 Configuring Skill Instance

2.5.1 Skill

The `Skill` object is the integration of all your skill logic. It is responsible for initializing SDK utilities such as the `AttributesManager` and `ServiceClientFactory` and also kick off the request handling process.

Available Methods

```
def invoke(self, request_envelope, context):
# type: (RequestEnvelope, Any) -> ResponseEnvelope
```

2.5.2 Skill Builders

The SDK includes a `SkillBuilder` that provides utility methods, to construct the `Skill` instance, setting custom user agent and creating lambda integration handler. It has the following structure:

```
class SkillBuilder(object):
    def __init__(self):
        # Initialize empty collections for request components,
        # exception handlers, interceptors.

    def add_request_handler(self, handler):
        # type: (AbstractRequestHandler) -> None
        ....

    def add_exception_handler(self, handler):
        # type: (AbstractExceptionHandler) -> None
        ....

    def add_global_request_interceptor(self, interceptor):
        # type: (AbstractRequestInterceptor) -> None
        ....

    def add_global_response_interceptor(self, interceptor):
        # type: (AbstractResponseInterceptor) -> None
        ....

    @property
    def skill_configuration(self):
        # type: () -> SkillConfiguration
        # Build configuration object using the registered components
        ....

    def create(self):
        # type: () -> Skill
        # Create the skill using the skill configuration
        ....

    def lambda_handler(self):
        # type: () -> LambdaHandler
        # Create a lambda handler function that can be tagged to
        # AWS Lambda handler.
```

(continues on next page)

(continued from previous page)

```
        # Processes the alexa request before invoking the skill,
        # processes the alexa response before providing to the service
        ....

    def request_handler(self, can_handle_func):
        # type: (Callable[[HandlerInput], bool]) -> None
        # Request Handler decorator

    def exception_handler(self, can_handle_func):
        # type: (Callable[[HandlerInput, Exception], bool]) -> None
        # Exception Handler decorator

    def global_request_interceptor(self):
        # type: () -> None
        # Global Request Interceptor decorator

    def global_response_interceptor(self):
        # type: () -> None
        # Global Response Interceptor decorator
```

There are two extensions to `SkillBuilder` class, `CustomSkillBuilder` and `StandardSkillBuilder`.

CustomSkillBuilder Class

`CustomSkillBuilder` is available in both `ask-sdk-core` and `ask-sdk` package. In addition to the common helper function above, `CustomSkillBuilder` also provides functions that allows you to register custom implementations of `AbstractPersistentAdapter` and `ask_sdk_model.services.ApiClient` classes.

```
class CustomSkillBuilder(SkillBuilder):
    def __init__(self, persistence_adapter=None, api_client=None):
        # type: (AbstractPersistenceAdapter, ApiClient) -> None
        ....

    @property
    def skill_configuration(self):
        # Create skill configuration from skill builder along with
        # registered persistence adapter and api client
        ....
```

StandardSkillBuilder Class

`StandardSkillBuilder` is available only in the `ask-sdk` package. It is a wrapper on CustomSKillBuilder with persistence_adapter as `ask_sdk_dynamo.adapter.DynamoDbPersistenceAdapter` and api_client as `ask_sdk_core.api_client.DefaultApiClient` to provide Persistence and Service Client features. It also provides optional parameters for configuring the Dynamo DB table options.

```
class StandardSkillBuilder(SkillBuilder):
    def __init__(self,
            table_name=None, auto_create_table=None,
            partition_keygen=None, dynamodb_client=None):
        # type: (str, bool, Callable[[RequestEnvelope], str], ServiceResource) ->␣
↪None)
        ....
```

(continues on next page)

(continued from previous page)

```
    @property
    def skill_configuration(self):
        # Create skill configuration from skill builder along with
        # default api client and dynamodb persistence adapter with
        # the passed in table configuration options.
        ....
```

2.6 Request Processing

Covers how to build request handlers, exception handlers, and request and response interceptors.

2.7 Response Building

Covers how to use the ResponseBuilder to compose multiple elements like text, cards, and audio into a single response.

2.8 Skill Attributes

Covers how to use skill attributes to store and retrieve skill data.

2.9 Alexa Service Clients

Covers how to use service clients in your skill to access Alexa APIs.

2.10 Skill Builders

Covers how to configure and construct a skill instance.

2.10.1 Runtime

Request Dispatch Components

Abstract Classes

class ask_sdk_runtime.dispatch.**AbstractRequestDispatcher**

Bases: object

Dispatcher which handles dispatching input request to the corresponding handler.

User needs to implement the dispatch method, to handle the processing of the incoming request in the handler input. A response may be expected out of the dispatch method.

dispatch(*handler_input*)

Dispatches an incoming request to the appropriate request handler and returns the output.

Parameters **handler_input** (*Input*) – generic input to the dispatcher

Returns generic output returned by handler in the dispatcher

Return type Union[None, Output]

class ask_sdk_runtime.dispatch_components.request_components.**AbstractRequestHandler**
Bases: object

Request Handlers are responsible for processing dispatch inputs and generating output.

Custom request handlers needs to implement `can_handle` and `handle` methods. `can_handle` returns True if the handler can handle the current input. `handle` processes the input and may return a output.

can_handle(*handler_input*)
Returns true if Request Handler can handle the dispatch input.

Parameters **handler_input** (*Input*) – Generic input passed to the dispatcher.

Returns Boolean value that tells the dispatcher if the current input can be handled by this handler.

Return type bool

handle(*handler_input*)
Handles the dispatch input and provides an output for dispatcher to return.

Parameters **handler_input** (*Input*) – Generic input passed to the dispatcher.

Returns Generic Output for the dispatcher to return or None

Return type Union[Output, None]

class ask_sdk_runtime.dispatch_components.request_components.**AbstractRequestInterceptor**
Bases: object

Interceptor that runs before the handler is called.

The `process` method has to be implemented, to run custom logic on the input, before it is handled by the Handler.

process(*handler_input*)
Process the input before the Handler is run.

Parameters **handler_input** (*Input*) – Generic input passed to the dispatcher.

Return type None

class ask_sdk_runtime.dispatch_components.request_components.**AbstractResponseInterceptor**
Bases: object

Interceptor that runs after the handler is called.

The `process` method has to be implemented, to run custom logic on the input and the dispatch output generated after the handler is executed on the input.

process(*handler_input*, *dispatch_output*)
Process the input and the output after the Handler is run.

Parameters

- **handler_input** (*Input*) – Generic input passed to the dispatcher.
- **dispatch_output** (*Union[None, Output]*) – Execution result of the Handler on dispatch input.

Return type None

class ask_sdk_runtime.dispatch_components.request_components.**AbstractRequestHandlerChain**

Bases: object

Abstract class containing Request Handler and corresponding Interceptors.

request_handler()

Returns Registered Request Handler instance.

Return type object

request_interceptors()

Returns List of registered Request Interceptors.

Return type list(*ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor*)

response_interceptors()

Returns List of registered Response Interceptors.

Return type list(*ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor*)

class ask_sdk_runtime.dispatch_components.request_components.**AbstractRequestMapper**

Bases: object

Class for request routing to the appropriate handler chain.

User needs to implement get_request_handler_chain method, to provide a routing mechanism of the input to the appropriate request handler chain containing the handler and the interceptors.

get_request_handler_chain(*handler_input*)

Get the handler chain that can process the handler input.

Parameters **handler_input** (*Input*) – Generic input passed to the dispatcher.

Returns Handler Chain that can handle the request under dispatch input.

Return type *AbstractRequestHandlerChain*

class ask_sdk_runtime.dispatch_components.request_components.**AbstractHandlerAdapter**

Bases: object

Abstracts handling of a request for specific handler types.

supports(*handler*)

Returns true if adapter supports the handler.

This method checks if the adapter supports the handler execution. This is usually checked by the type of the handler.

Parameters **handler** (*object*) – Request Handler instance.

Returns Boolean denoting whether the adapter supports the handler.

Return type bool

execute(*handler_input*, *handler*)

Executes the handler with the provided dispatch input.

Parameters

- **handler_input** (*Input*) – Generic input passed to the dispatcher.
- **handler** (*object*) – Request Handler instance.

Returns Result executed by passing handler_input to handler.

Return type Union[None, Output]

class ask_sdk_runtime.dispatch_components.exception_components.**AbstractExceptionHandler**

Bases: object

Handles exception types and optionally produce an output.

The abstract class is similar to Request Handler, with methods can_handle and handle. The can_handle method checks if the handler can support the input and the exception. The handle method processes the input and exception, to optionally produce an output.

can_handle(*handler_input*, *exception*)

Checks if the handler can support the exception raised during dispatch.

Parameters

- **handler_input** (*Input*) – Generic input passed to the dispatcher.
- **exception** (*Exception*) – Exception raised during dispatch.

Returns Boolean whether handler can handle exception or not.

Return type bool

handle(*handler_input*, *exception*)

Process the dispatch input and exception.

Parameters

- **handler_input** (*Input*) – Generic input passed to the dispatcher.
- **exception** (*Exception*) – Exception raised during dispatch.

Returns Optional output object to serve as dispatch return.

Return type Union[None, Output]

class ask_sdk_runtime.dispatch_components.exception_components.**AbstractExceptionMapper**

Bases: object

Mapper to register custom Exception Handler instances.

The exception mapper is used by *ask_sdk_runtime.dispatch.GenericRequestDispatcher* dispatch method, to handle exceptions. The mapper can contain one or more exception handlers. Handlers are accessed through the mapper to attempt to find a handler that is compatible with the current exception.

get_handler(*handler_input*, *exception*)

Returns a suitable exception handler to dispatch the specified exception, if one exists.

Parameters

- **handler_input** (*Input*) – Generic input passed to the dispatcher.
- **exception** (*Exception*) – Exception thrown by *ask_sdk_runtime.dispatch.GenericRequestDispatcher* dispatch method.

Returns Exception Handler that can handle the input or None.

Return type Union[None, *AbstractExceptionHandler*]

Implementations

class ask_sdk_runtime.dispatch.**GenericRequestDispatcher**(*options*)

Bases: *ask_sdk_runtime.dispatch.AbstractRequestDispatcher*

Generic implementation of *AbstractRequestDispatcher*.

The runtime configuration contains the components required for the dispatcher, which is passed during initialization.

When the dispatch method is invoked, using a list of `ask_sdk_runtime.dispatch_components.request_components.RequestMapper` , the Dispatcher finds a handler for the request and delegates the invocation to the supported `ask_sdk_runtime.dispatch_components.request_components.HandlerAdapter` . If the handler raises any exception, it is delegated to `ask_sdk_runtime.dispatch_components.exception_components.ExceptionMapper` to handle or raise it to the upper stack.

`dispatch`(*handler_input*)

Dispatches an incoming request to the appropriate request handler and returns the output.

Before running the request on the appropriate request handler, dispatcher runs any predefined global request interceptors. On successful response returned from request handler, dispatcher runs predefined global response interceptors, before returning the response.

Parameters **`handler_input`** (*`Input`*) – generic input to the dispatcher

Returns generic output handled by the handler, optionally containing a response

Return type Union[None, Output]

Raises *`ask_sdk_runtime.exceptions.DispatchException`*

class `ask_sdk_runtime.dispatch_components.request_components.`**`GenericRequestHandlerChain`**(*req reque respo*

Bases: *`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandlerChain`*

Generic implementation of *`AbstractRequestHandlerChain`*.

Generic Request Handler Chain accepts request handler of any type.

Parameters

- **`request_handler`** (`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler`) – Registered Request Handler instance of generic type.
- **`request_interceptors`** (*`list`*(`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor`*)*) – List of registered Request Interceptors.
- **`response_interceptors`** (*`list`*(`ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor`*)*) – List of registered Response Interceptors.

`request_handler`

Returns Registered Request Handler instance.

Return type object

`request_interceptors`

Returns List of registered Request Interceptors.

Return type list(*ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor*)

`response_interceptors`

Returns List of registered Response Interceptors.

Return type list(*ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor*)

`add_request_interceptor`(*interceptor*)

Add interceptor to Request Interceptors list.

Parameters **`interceptor`** (`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor`) – Request Interceptor instance.

`add_response_interceptor`(*interceptor*)

Add interceptor to Response Interceptors list.

Parameters **`interceptor`** (`ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor`) – Response Interceptor instance.

class `ask_sdk_runtime.dispatch_components.request_components.`**`GenericRequestMapper`**(*request_hand*

Bases: *`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestMapper`*

Implementation of *`AbstractRequestMapper`* that registers `RequestHandlerChain`.

The class accepts request handler chains of type *`GenericRequestHandlerChain`* only. The `get_request_handler_chain` method returns the *`GenericRequestHandlerChain`* instance that can handle the request in the handler input.

Parameters **`request_handler_chains`** (*`list`*(`GenericRequestHandlerChain`)) – List of *`GenericRequestHandlerChain`* instances.

`request_handler_chains`

Returns List of *`GenericRequestHandlerChain`* instances.

Return type list(*GenericRequestHandlerChain*)

`add_request_handler_chain`(*request_handler_chain*)

Checks the type before adding it to the request_handler_chains instance variable.

Parameters **`request_handler_chain`** (*`RequestHandlerChain`*) – Request Handler Chain instance.

Raises *`ask_sdk_runtime.exceptions.DispatchException`* if a null input is provided or if the input is of invalid type

`get_request_handler_chain`(*handler_input*)

Get the request handler chain that can handle the dispatch input.

Parameters **`handler_input`** (*`Input`*) – Generic input passed to the dispatcher.

Returns Handler Chain that can handle the input.

Return type Union[None, *GenericRequestHandlerChain*]

class `ask_sdk_runtime.dispatch_components.request_components.`**`GenericHandlerAdapter`**

Bases: *`ask_sdk_runtime.dispatch_components.request_components.AbstractHandlerAdapter`*

GenericHandler Adapter for handlers of type *`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler`*.

`supports`(*handler*)

Returns true if handler is *`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler`* instance.

Parameters **handler** (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler) – Request Handler instance

Returns Boolean denoting whether the adapter supports the handler.

Return type bool

execute (*handler_input*, *handler*)
Executes the handler with the provided handler input.

Parameters

- **handler_input** (*Input*) – Generic input passed to the dispatcher.
- **handler** (*object*) – Request Handler instance.

Returns Result executed by passing handler_input to handler.

Return type Union[None, Output]

class ask_sdk_runtime.dispatch_components.exception_components.**GenericExceptionMapper** (*except*
Bases: *ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionMapper*

Generic Implementation of exception mapper, to register *AbstractExceptionHandler* instances.

The class accepts exception handlers of type *AbstractExceptionHandler* only. The get_handler method returns the *AbstractExceptionHandler* instance that can handle the dispatch input and the exception raised from the dispatch method.

Parameters **exception_handlers** (*list* (ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler*)*) – List of *ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler* instances.

exception_handlers

Returns List of *ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler* instances.

Return type list(*ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler*)

add_exception_handler (*exception_handler*)
Checks the type before adding it to the exception_handlers instance variable.

Parameters **exception_handler** (ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler) – Exception Handler instance.

Raises *ask_sdk_runtime.exceptions.DispatchException* if a null input is provided or if the input is of invalid type

get_handler (*handler_input*, *exception*)
Get the exception handler that can handle the input and exception.

Parameters

- **handler_input** (*Input*) – Generic input passed to the dispatcher.
- **exception** (*Exception*) – Exception thrown by *ask_sdk_runtime.dispatch.GenericRequestDispatcher* dispatch method.

Returns Exception Handler that can handle the input or None.

Return type Union[None, *ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler*]

Skill Components

class ask_sdk_runtime.skill.**RuntimeConfiguration**(*request_mappers*, *handler_adapters*, *request_interceptors=None*, *response_interceptors=None*, *exception_mapper=None*)

Bases: `object`

Configuration Object that represents standard components needed to build the dispatcher in the *AbstractSkill*.

Parameters

- **request_mappers** (*list* (GenericRequestMapper)) – List of request mapper instances.
- **handler_adapters** (*list* (GenericHandlerAdapter)) – List of handler adapter instances.
- **request_interceptors** (*list* (AbstractRequestInterceptor)) – List of request interceptor instances.
- **response_interceptors** (*list* (AbstractResponseInterceptor)) – List of response interceptor instances.
- **exception_mapper** (GenericExceptionMapper) – Exception mapper instance.

class ask_sdk_runtime.skill.**RuntimeConfigurationBuilder**

Bases: `object`

Builder class for creating a runtime configuration object, from base dispatch components.

add_request_handler(*request_handler*)

Register input to the request handlers list.

Parameters **request_handler** (AbstractRequestHandler) – Request Handler instance to be registered.

Returns None

add_request_handlers(*request_handlers*)

Register input to the request handlers list.

Parameters **request_handlers** (*list* (AbstractRequestHandler)) – List of Request Handler instances to be registered.

Returns None

add_exception_handler(*exception_handler*)

Register input to the exception handlers list.

Parameters **exception_handler** (AbstractExceptionHandler) – Exception Handler instance to be registered.

Returns None

add_global_request_interceptor(*request_interceptor*)

Register input to the global request interceptors list.

Parameters **request_interceptor** (AbstractRequestInterceptor) – Request Interceptor instance to be registered.

Returns None

add_global_response_interceptor(*response_interceptor*)
Register input to the global response interceptors list.

Parameters **response_interceptor** (AbstractResponseInterceptor) – Response Interceptor instance to be registered.

Returns None

get_runtime_configuration()
Build the runtime configuration object from the registered components.

Returns Runtime Configuration Object

Return type *RuntimeConfiguration*

class ask_sdk_runtime.skill.**AbstractSkill**
Bases: object

Abstract class that acts as entry level container for skill invocation.

Domain SDKs should implement the *supports* and *invoke* methods.

supports(*event*, *context*)
Check if the skill supports the corresponding input.

Parameters

- **event** (*SkillInput*) – input instance containing request information.
- **context** (*Any*) – Context passed during invocation

Returns boolean if this type of request can be handled by this skill.

Return type bool

invoke(*event*, *context*)
Invokes the dispatcher, to handle the skill input and return a skill output.

Parameters

- **event** (*SkillInput*) – input instance containing request information.
- **context** (*Any*) – Context passed during invocation

Returns output generated by handling the request.

Return type SkillOutput

class ask_sdk_runtime.skill_builder.**AbstractSkillBuilder**
Bases: object

Abstract Skill Builder with helper functions for building *ask_sdk_runtime.skill.AbstractSkill* object.

Domain SDKs has to implement the *create* method that returns an instance of the skill implementation for the domain type.

add_request_handler(*request_handler*)
Register input to the request handlers list.

Parameters **request_handler** (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler) – Request Handler instance to be registered.

Returns None

add_exception_handler(*exception_handler*)

Register input to the exception handlers list.

Parameters **exception_handler** (*ask_sdk_runtime.dispatch_components.request_components.AbstractExceptionHandler*) – Exception Handler instance to be registered.

Returns None

add_global_request_interceptor(*request_interceptor*)

Register input to the global request interceptors list.

Parameters **request_interceptor** (`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor`) – Request Interceptor instance to be registered.

Returns None

add_global_response_interceptor(*response_interceptor*)

Register input to the global response interceptors list.

Parameters **response_interceptor** (`ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor`) – Response Interceptor instance to be registered.

Returns None

request_handler(*can_handle_func*)

Decorator that can be used to add request handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes a single parameter and no varargs or kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that returns a response object by the skill. The function should follow the signature of the handle function in *ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler* class.

Parameters **can_handle_func** (*Callable[[Input], bool]*) – The function that validates if the request can be handled.

Returns Wrapper function that can be decorated on a handle function.

exception_handler(*can_handle_func*)

Decorator that can be used to add exception handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes two parameters and no varargs or kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that processes the exception raised during dispatcher and returns a response object by the skill. The function should follow the signature of the handle function in *ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler* class.

Parameters **can_handle_func** (*Callable[[Input, Exception], bool]*) – The function that validates if the exception can be handled.

Returns Wrapper function that can be decorated on a handle function.

global_request_interceptor()

Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input. The function should follow the signature of the process function in *ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor* class.

Returns Wrapper function that can be decorated on a interceptor process function.

global_response_interceptor()

Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input and the response generated by the request handler. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor` class.

Returns Wrapper function that can be decorated on a interceptor process function.

create()

Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type *AbstractSkill*

SDK Exceptions

exception ask_sdk_runtime.exceptions.**AskSdkException**

Bases: `Exception`

Base class for exceptions raised by the SDK.

exception ask_sdk_runtime.exceptions.**DispatchException**

Bases: `ask_sdk_runtime.exceptions.AskSdkException`

Class for exceptions raised during dispatch logic.

exception ask_sdk_runtime.exceptions.**SerializationException**

Bases: `ask_sdk_runtime.exceptions.AskSdkException`

Class for exceptions raised during serialization/deserialization.

exception ask_sdk_runtime.exceptions.**SkillBuilderException**

Bases: `ask_sdk_runtime.exceptions.AskSdkException`

Base exception class for Skill Builder exceptions.

exception ask_sdk_runtime.exceptions.**RuntimeConfigException**

Bases: `ask_sdk_runtime.exceptions.AskSdkException`

Base exception class for Runtime Configuration Builder exceptions.

2.10.2 Core

Handler Input

class ask_sdk_core.handler_input.**HandlerInput**(*request_envelope*, *attributes_manager=None*, *context=None*, *service_client_factory=None*)

Bases: `object`

Input to Request Handler, Exception Handler and Interceptors.

Handler Input instantiations are passed to the registered instances of *AbstractRequestHandler* and *AbstractExceptionHandler* , during skill invocation. The class provides a *AttributesManager* and a *ResponseFactory* instance, apart from *RequestEnvelope*, *Context* and *ServiceClientFactory* instances, to utilize during the lifecycle of skill.

Parameters

- **request_envelope** (ask_sdk_model.request_envelope.RequestEnvelope) – Request Envelope passed from Alexa Service
- **attributes_manager** (ask_sdk_core.attributes_manager.AttributesManager) – Attribute Manager instance for managing attributes across skill lifecycle
- **context** (*object*) – Context object passed from Lambda service
- **service_client_factory** (ask_sdk_model.services.service_client_factory.ServiceClientFactory) – Service Client Factory instance for calling Alexa services

service_client_factory

Service Client Factory instance for calling Alexa services.

To use the Alexa services, one need to configure the API Client in the skill builder object, before creating the skill.

Request Dispatch Components

Abstract Classes

class ask_sdk_core.dispatch_components.request_components.**AbstractRequestHandler**

Bases: *ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler*

Request Handlers are responsible for processing Request inside the Handler Input and generating Response.

Custom request handlers needs to implement `can_handle` and `handle` methods. `can_handle` returns True if the handler can handle the current request. `handle` processes the Request and may return a Response.

can_handle(*handler_input*)

Returns true if Request Handler can handle the Request inside Handler Input.

Parameters **handler_input** (HandlerInput) – Handler Input instance with Request Envelope containing Request.

Returns Boolean value that tells the dispatcher if the current request can be handled by this handler.

Return type bool

handle(*handler_input*)

Handles the Request inside handler input and provides a Response for dispatcher to return.

Parameters **handler_input** (HandlerInput) – Handler Input instance with Request Envelope containing Request.

Returns Response for the dispatcher to return or None

Return type Union[*Response*, None]

class ask_sdk_core.dispatch_components.request_components.**AbstractRequestInterceptor**

Bases: *ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor*

Interceptor that runs before the handler is called.

The `process` method has to be implemented, to run custom logic on the input, before it is handled by the Handler.

process(*handler_input*)

Process the input before the Handler is run.

Parameters **handler_input** (`HandlerInput`) – Handler Input instance.

Return type None

class ask_sdk_core.dispatch_components.request_components.**AbstractResponseInterceptor**

Bases: *ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor*

Interceptor that runs after the handler is called.

The `process` method has to be implemented, to run custom logic on the input and the response generated after the handler is executed on the input.

process(*handler_input*, *response*)

Process the input and the response after the Handler is run.

Parameters

- **handler_input** (`HandlerInput`) – Handler Input instance.
- **response** (Union[None, `ask_sdk_model.Response`]) – Execution result of the Handler on handler input.

Return type None

class ask_sdk_core.dispatch_components.exception_components.**AbstractExceptionHandler**

Bases: *ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler*

Handles exception types and optionally produce a response.

The abstract class is similar to Request Handler, with methods can_handle and handle. The `can_handle` method checks if the handler can support the input and the exception. The `handle` method processes the input and exception, to optionally produce a response.

can_handle(*handler_input*, *exception*)

Checks if the handler can support the exception raised during dispatch.

Parameters

- **handler_input** (`HandlerInput`) – Handler Input instance.
- **exception** (*Exception*) – Exception raised during dispatch.

Returns Boolean whether handler can handle exception or not.

Return type bool

handle(*handler_input*, *exception*)

Process the handler input and exception.

Parameters

- **handler_input** (`HandlerInput`) – Handler Input instance.
- **exception** (*Exception*) – Exception raised during dispatch.

Returns Optional response object to serve as dispatch return.

Return type Union[None, *Response*]

Response Builder Components

`ask_sdk_core.response_helper.`**`PLAIN_TEXT_TYPE = 'PlainText'`**

str: Helper variable for plain text type.

`ask_sdk_core.response_helper.`**`RICH_TEXT_TYPE = 'RichText'`**

str: Helper variable for rich text type.

`class` `ask_sdk_core.response_helper.`**`ResponseFactory`**

Bases: `object`

ResponseFactory is class which provides helper functions to help building a response.

`speak`(*speech*)

Say the provided speech to the user.

Parameters **`speech`** (`str`) – the output speech sent back to the user.

Returns response factory with partial response being built and access from self.response.

Return type *ResponseFactory*

`ask`(*reprompt*)

Provide reprompt speech to the user, if no response for 8 seconds.

The should_end_session value will be set to false except when the video app launch directive is present in directives.

Parameters **`reprompt`** (`str`) – the output speech to reprompt.

Returns response factory with partial response being built and access from self.response.

Return type *ResponseFactory*

`set_card`(*card*)

Renders a card within the response.

For more information about card object in response, click here: https://developer.amazon.com/docs/custom-skills/request-and-response-json-reference.html#card-object.

Parameters **`card`** (`ask_sdk_model.ui.card.Card`) – card object in response sent back to user.

Returns response factory with partial response being built and access from self.response.

Return type *ResponseFactory*

`add_directive`(*directive*)

Adds directive to response.

Parameters **`directive`** (`ask_sdk_model.directive.Directive`) – the directive sent back to Alexa device.

Returns response factory with partial response being built and access from self.response.

Return type *ResponseFactory*

`set_should_end_session`(*should_end_session*)

Sets shouldEndSession value to null/false/true.

Parameters **`should_end_session`** (`bool`) – value to show if the session should be ended or not.

Returns response factory with partial response being built and access from self.response.

Return type *ResponseFactory*

set_can_fulfill_intent(*can_fulfill_intent*)

Sets CanFulfill intent to the response.

For more information on CanFulfillIntent, check the name-free interaction doc here: https://developer.amazon.com/docs/custom-skills/understand-name-free-interaction-for-custom-skills.html

Parameters **can_fulfill_intent** (`CanFulfillIntent`) – CanFulfill Intent sent back in response.

Returns response factory with partial response being built and access from self.response.

Return type *ResponseFactory*

`ask_sdk_core.response_helper.`**get_plain_text_content**(*primary_text=None, secondary_text=None, tertiary_text=None*)

Responsible for building plain text content object using ask-sdk-model in Alexa skills kit display interface. https://developer.amazon.com/docs/custom-skills/display-interface-reference.html#textcontent-object-specifications.

Parameters

- **primary_text** (*(optional) str*) – Text for primary_text field
- **secondary_text** (*(optional) str*) – Text for secondary_text field
- **tertiary_text** (*(optional) str*) – Text for tertiary_text field

Returns Text Content instance with primary, secondary and tertiary text set as Plain Text objects.

Return type *TextContent*

Raises ValueError

`ask_sdk_core.response_helper.`**get_rich_text_content**(*primary_text=None, secondary_text=None, tertiary_text=None*)

Responsible for building plain text content object using ask-sdk-model in Alexa skills kit display interface. https://developer.amazon.com/docs/custom-skills/display-interface-reference.html#textcontent-object-specifications.

Parameters

- **primary_text** (*(optional) str*) – Text for primary_text field
- **secondary_text** (*(optional) str*) – Text for secondary_text field
- **tertiary_text** (*(optional) str*) – Text for tertiary_text field

Returns Text Content instance with primary, secondary and tertiary text set as Rich Text objects.

Return type *TextContent*

Raises ValueError

`ask_sdk_core.response_helper.`**get_text_content**(*primary_text=None, primary_text_type='PlainText', secondary_text=None, secondary_text_type='PlainText', tertiary_text=None, tertiary_text_type='PlainText'*)

Responsible for building text content object using ask-sdk-model in Alexa skills kit display interface. https://developer.amazon.com/docs/custom-skills/display-interface-reference.html#textcontent-object-specifications.

Parameters

- **primary_text** (*(optional)* *str*) – Text for primary_text field
- **primary_text_type** (*(optional)* *str*) – Type of the primary text field. Allowed values are *PlainText* and *RichText*. Defaulted to *PlainText*.
- **secondary_text** (*(optional)* *str*) – Text for secondary_text field
- **secondary_text_type** – Type of the secondary text field. Allowed values are *PlainText* and *RichText*. Defaulted to *PlainText*.
- **tertiary_text** (*(optional)* *str*) – Text for tertiary_text field
- **tertiary_text_type** – Type of the tertiary text field. Allowed values are *PlainText* and *RichText*. Defaulted to *PlainText*.

Returns Text Content instance with primary, secondary and tertiary text set.

Return type *TextContent*

Raises ValueError

Skill Components

class ask_sdk_core.skill.**SkillConfiguration**(*request_mappers*, *handler_adapters*, *request_interceptors=None*, *response_interceptors=None*, *exception_mapper=None*, *persistence_adapter=None*, *api_client=None*, *custom_user_agent=None*, *skill_id=None*)

Bases: *ask_sdk_runtime.skill.RuntimeConfiguration*

Configuration Object that represents standard components needed to build `Skill`.

Parameters

- **request_mappers** (*list* (GenericRequestMapper)) – List of request mapper instances.
- **handler_adapters** (*list* (GenericHandlerAdapter)) – List of handler adapter instances.
- **request_interceptors** (*list* (ask_sdk_core.dispatch_components.request_components.AbstractRequestInterceptor)) – List of request interceptor instances.
- **response_interceptors** (*list* (ask_sdk_core.dispatch_components.request_components.AbstractResponseInterceptor)) – List of response interceptor instances.
- **exception_mapper** (GenericExceptionMapper) – Exception mapper instance.
- **persistence_adapter** (AbstractPersistenceAdapter) – Persistence adapter instance.
- **api_client** (ask_sdk_model.services.api_client.ApiClient) – Api Client instance.
- **custom_user_agent** (*str*) – Custom User Agent string
- **skill_id** (*str*) – ID of the skill.

class ask_sdk_core.skill.**CustomSkill**(*skill_configuration*)

Bases: *ask_sdk_runtime.skill.AbstractSkill*

Top level container for Request Dispatcher, Persistence Adapter and Api Client.

Parameters **skill_configuration** (`SkillConfiguration`) – Configuration object that holds information about different components needed to build the skill object.

supports (*request_envelope*, *context*)
Check if request envelope is of the expected skill format.

Parameters

- **request_envelope** (*SkillInput*) – input instance containing request information.
- **context** (*Any*) – Context passed during invocation

Returns boolean if this type of request can be handled by this skill.

Return type bool

invoke (*request_envelope*, *context*)
Invokes the dispatcher, to handle the request envelope and return a response envelope.

Parameters

- **request_envelope** (`RequestEnvelope`) – Request Envelope instance containing request information
- **context** (*Any*) – Context passed during invocation

Returns Response Envelope generated by handling the request

Return type *ResponseEnvelope*

class ask_sdk_core.skill_builder.**SkillBuilder**
Bases: *ask_sdk_runtime.skill_builder.AbstractSkillBuilder*

Skill Builder with helper functions for building `ask_sdk_core.skill.Skill` object.

skill_configuration
Create the skill configuration object using the registered components.

create ()
Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type Skill

lambda_handler ()
Create a handler function that can be used as handler in AWS Lambda console.

The lambda handler provides a handler function, that acts as an entry point to the AWS Lambda console. Users can set the lambda_handler output to a variable and set the variable as AWS Lambda Handler on the console.

Returns Handler function to tag on AWS Lambda console.

add_exception_handler (*exception_handler*)
Register input to the exception handlers list.

Parameters **exception_handler** (*ask_sdk_runtime.dispatch_components.request_components.AbstractExceptionHandler*) – Exception Handler instance to be registered.

Returns None

add_global_request_interceptor (*request_interceptor*)
Register input to the global request interceptors list.

Parameters **request_interceptor** (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor) – Request Interceptor instance to be registered.

Returns None

add_global_response_interceptor(*response_interceptor*)
Register input to the global response interceptors list.

Parameters **response_interceptor** (ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor) – Response Interceptor instance to be registered.

Returns None

add_request_handler(*request_handler*)
Register input to the request handlers list.

Parameters **request_handler** (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler) – Request Handler instance to be registered.

Returns None

exception_handler(*can_handle_func*)
Decorator that can be used to add exception handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes two parameters and no varargs or kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that processes the exception raised during dispatcher and returns a response object by the skill. The function should follow the signature of the handle function in *ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler* class.

Parameters **can_handle_func** (*Callable[[Input, Exception], bool]*) – The function that validates if the exception can be handled.

Returns Wrapper function that can be decorated on a handle function.

global_request_interceptor()
Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input. The function should follow the signature of the process function in *ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor* class.

Returns Wrapper function that can be decorated on a interceptor process function.

global_response_interceptor()
Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input and the response generated by the request handler. The function should follow the signature of the process function in *ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor* class.

Returns Wrapper function that can be decorated on a interceptor process function.

request_handler(*can_handle_func*)
Decorator that can be used to add request handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes a single parameter and no varargs or kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that returns a response object by the skill. The function should follow the signature of the handle function in *ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler* class.

Parameters **can_handle_func** (*Callable[[Input], bool]*) – The function that validates if the request can be handled.

Returns Wrapper function that can be decorated on a handle function.

class ask_sdk_core.skill_builder.**CustomSkillBuilder**(*persistence_adapter=None, api_client=None*)

Bases: *ask_sdk_core.skill_builder.SkillBuilder*

Skill Builder with api client and persistence adapter setter functions.

skill_configuration

Create the skill configuration object using the registered components.

add_exception_handler(*exception_handler*)

Register input to the exception handlers list.

Parameters **exception_handler** (*ask_sdk_runtime.dispatch_components.request_components.AbstractExceptionHandler*) – Exception Handler instance to be registered.

Returns None

add_global_request_interceptor(*request_interceptor*)

Register input to the global request interceptors list.

Parameters **request_interceptor** (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor) – Request Interceptor instance to be registered.

Returns None

add_global_response_interceptor(*response_interceptor*)

Register input to the global response interceptors list.

Parameters **response_interceptor** (ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor) – Response Interceptor instance to be registered.

Returns None

add_request_handler(*request_handler*)

Register input to the request handlers list.

Parameters **request_handler** (ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler) – Request Handler instance to be registered.

Returns None

create()

Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type Skill

exception_handler(*can_handle_func*)

Decorator that can be used to add exception handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes two parameters and no varargs or kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that processes the exception raised during dispatcher and returns a response object by the skill. The function should follow the signature of the handle function in `ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler` class.

Parameters **`can_handle_func`** (`Callable[[Input, Exception], bool]`) – The function that validates if the exception can be handled.

Returns Wrapper function that can be decorated on a handle function.

`global_request_interceptor``()`

Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor` class.

Returns Wrapper function that can be decorated on a interceptor process function.

`global_response_interceptor``()`

Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input and the response generated by the request handler. The function should follow the signature of the process function in `ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor` class.

Returns Wrapper function that can be decorated on a interceptor process function.

`lambda_handler``()`

Create a handler function that can be used as handler in AWS Lambda console.

The lambda handler provides a handler function, that acts as an entry point to the AWS Lambda console. Users can set the lambda_handler output to a variable and set the variable as AWS Lambda Handler on the console.

Returns Handler function to tag on AWS Lambda console.

`request_handler``(`*can_handle_func*`)`

Decorator that can be used to add request handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes a single parameter and no varargs or kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that returns a response object by the skill. The function should follow the signature of the handle function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler` class.

Parameters **`can_handle_func`** (`Callable[[Input], bool]`) – The function that validates if the request can be handled.

Returns Wrapper function that can be decorated on a handle function.

Skill Attribute Components

`class` `ask_sdk_core.attributes_manager.`**`AttributesManager`**`(`*request_envelope*, *persistence_adapter=None*`)`

Bases: `object`

AttributesManager is a class that handles three level attributes: request, session and persistence.

Parameters

- **request_envelope** (RequestEnvelope) – request envelope.
- **persistence_adapter** (AbstractPersistenceAdapter) – class used for storing and retrieving persistent attributes from persistence tier

request_attributes

Attributes stored at the Request level of the skill lifecycle.

Returns request attributes for the request life cycle

Return type Dict[str, object]

session_attributes

Attributes stored at the Session level of the skill lifecycle.

Returns session attributes extracted from request envelope

Return type Dict[str, object]

persistent_attributes

Attributes stored at the Persistence level of the skill lifecycle.

Returns persistent_attributes retrieved from persistence adapter

Return type Dict[str, object]

Raises *ask_sdk_core.exceptions.AttributesManagerException* if trying to get persistent attributes without persistence adapter

save_persistent_attributes()

Save persistent attributes to the persistence layer if a persistence adapter is provided.

Return type None

Raises *ask_sdk_core.exceptions.AttributesManagerException* if trying to save persistence attributes without persistence adapter

Abstract Classes

class ask_sdk_core.attributes_manager.**AbstractPersistenceAdapter**

Bases: object

Abstract class for storing and retrieving persistent attributes from persistence tier given request envelope.

User needs to implement get_attributes method to get attributes from persistent tier and save_attributes method to save attributes to persistent tier.

get_attributes(*request_envelope*)

Get attributes from persistent tier.

Parameters **request_envelope** (RequestEnvelope) – Request Envelope from Alexa service

Returns A dictionary of attributes retrieved from persistent tier

Return type Dict[str, object]

save_attributes(*request_envelope*, *attributes*)

Save attributes to persistent tier.

Parameters

- **request_envelope** (RequestEnvelope) – request envelope.

- **attributes** (*Dict[str, object]*) – attributes to be saved to persistent tier

Return type None

API Client

class ask_sdk_core.api_client.**DefaultApiClient**

Bases: *ask_sdk_model.services.api_client.ApiClient*

Default ApiClient implementation of *ask_sdk_model.services.api_client.ApiClient* using the *requests* library.

invoke (*request*)

Dispatches a request to an API endpoint described in the request.

Resolves the method from input request object, converts the list of header tuples to the required format (dict) for the *requests* lib call and invokes the method with corresponding parameters on *requests* library. The response from the call is wrapped under the *ApiClientResponse* object and the responsibility of translating a response code and response/ error lies with the caller.

Parameters **request** (ApiClientRequest) – Request to dispatch to the ApiClient

Returns Response from the client call

Return type *ApiClientResponse*

Raises *ask_sdk_core.exceptions.ApiClientException*

SDK Exceptions

exception ask_sdk_core.exceptions.**AttributesManagerException**

Bases: *ask_sdk_runtime.exceptions.AskSdkException*

Class for exceptions raised during handling attributes logic

exception ask_sdk_core.exceptions.**SerializationException**

Bases: *ask_sdk_runtime.exceptions.AskSdkException*

Class for exceptions raised during serialization/deserialization.

exception ask_sdk_core.exceptions.**PersistenceException**

Bases: *ask_sdk_runtime.exceptions.AskSdkException*

Exception class for Persistence Adapter processing.

exception ask_sdk_core.exceptions.**ApiClientException**

Bases: *ask_sdk_runtime.exceptions.AskSdkException*

Exception class for ApiClient Adapter processing.

Default Serializer

class ask_sdk_core.serialize.**DefaultSerializer**

Bases: *ask_sdk_model.services.serializer.Serializer*

serialize (*obj*)

Builds a serialized object.

- If obj is None, return None.
- If obj is str, int, long, float, bool, return directly.

- If obj is datetime.datetime, datetime.date convert to string in iso8601 format.
- If obj is list, serialize each element in the list.
- If obj is dict, return the dict with serialized values.
- If obj is ask sdk model, return the dict with keys resolved from the union of model's `attribute_map` and `deserialized_types` and values serialized based on `deserialized_types`.
- If obj is a generic class instance, return the dict with keys from instance's `deserialized_types` and values serialized based on `deserialized_types`.

Parameters **obj** (*object*) – The data to serialize.

Returns The serialized form of data.

Return type Union[Dict[str, Any], List[Any], Tuple[Any], str, None]

deserialize(*payload*, *obj_type*)

Deserializes payload into an instance of provided `obj_type`.

The `obj_type` parameter can be a primitive type, a generic model object or a list / dict of model objects.

The list or dict object type has to be provided as a string format. For eg:

- `'list[a.b.C]'` if the payload is a list of instances of class `a.b.C`.
- `'dict(str, a.b.C)'` if the payload is a dict containing mappings of `str : a.b.C` class instance types.

The method looks for a `deserialized_types` dict in the model class, that mentions which payload values has to be deserialized. In case the payload key names are different than the model attribute names, the corresponding mapping can be provided in another special dict `attribute_map`. The model class should also have the __init__ method with default values for arguments. Check *ask_sdk_model.request_envelope.RequestEnvelope* source code for an example implementation.

Parameters

- **payload** (*str*) – data to be deserialized.
- **obj_type** (*Union[str, object]*) – resolved class name for deserialized object

Returns deserialized object

Return type object

Raises *ask_sdk_core.exceptions.SerializationException*

General Utilities

2.10.3 DynamoDb Persistence Adapter

DynamoDb Persistence Adapter

class ask_sdk_dynamodb.adapter.**DynamoDbAdapter**(*table_name*, *partition_key_name='id'*, *attribute_name='attributes'*, *create_table=False*, *partition_keygen=<function user_id_partition_keygen>*, *dynamodb_resource=dynamodb.ServiceResource()*)

Bases: *ask_sdk_core.attributes_manager.AbstractPersistenceAdapter*

Persistence Adapter implementation using Amazon DynamoDb.

Amazon DynamoDb based persistence adapter implementation. This internally uses the AWS Python SDK (*boto3*) to process the dynamodb operations. The adapter tries to create the table if `create_table` is set, during initialization.

Parameters

- **`table_name`** (*`str`*) – Name of the table to be created or used
- **`partition_key_name`** (*`str`*) – Partition key name to be used. Defaulted to 'id'
- **`attribute_name`** (*`str`*) – Attribute name for storing and retrieving attributes from dynamodb. Defaulted to 'attributes'
- **`create_table`** (*`bool`*) – Should the adapter try to create the table if it doesn't exist. Defaulted to False
- **`partition_keygen`** (*`Callable[[RequestEnvelope], str]`*) – Callable function that takes a request envelope and provides a unique partition key value. Defaulted to user id keygen function
- **`dynamodb_resource`** (*`boto3.resources.base.ServiceResource`*) – Resource to be used, to perform dynamo operations. Defaulted to resource generated from boto3

`get_attributes`(*request_envelope*)

Get attributes from table in Dynamodb resource.

Retrieves the attributes from Dynamodb table. If the table doesn't exist, returns an empty dict if the `create_table` variable is set as True, else it raises PersistenceException. Raises PersistenceException if *get_item* fails on the table.

Parameters **`request_envelope`** (*`ask_sdk_model.RequestEnvelope`*) – Request Envelope passed during skill invocation

Returns Attributes stored under the partition keygen mapping in the table

Return type Dict[str, object]

Raises *`ask_sdk_core.exceptions.PersistenceException`*

`save_attributes`(*request_envelope*, *attributes*)

Saves attributes to table in Dynamodb resource.

Saves the attributes into Dynamodb table. Raises PersistenceException if table doesn't exist or `put_item` fails on the table.

Parameters

- **`request_envelope`** (*`ask_sdk_model.RequestEnvelope`*) – Request Envelope passed during skill invocation
- **`attributes`** (*`Dict[str, object]`*) – Attributes stored under the partition keygen mapping in the table

Return type None

Raises *`ask_sdk_core.exceptions.PersistenceException`*

Partition Key Generator Functions

`ask_sdk_dynamodb.partition_keygen.`**`device_id_partition_keygen`**(*request_envelope*)

Retrieve device id from request envelope, to use as partition key.

Parameters **request_envelope** (*ask_sdk_model.RequestEnvelope*) – Request Envelope passed during skill invocation

Returns Device Id retrieved from request envelope

Return type str

Raises *ask_sdk_core.exceptions.PersistenceException*

ask_sdk_dynamodb.partition_keygen.**user_id_partition_keygen**(*request_envelope*)
Retrieve user id from request envelope, to use as partition key.

Parameters **request_envelope** (*ask_sdk_model.RequestEnvelope*) – Request Envelope passed during skill invocation

Returns User Id retrieved from request envelope

Return type str

Raises *ask_sdk_core.exceptions.PersistenceException*

2.10.4 Standard

Standard Skill Builder

class ask_sdk.standard.**StandardSkillBuilder**(*table_name=None*, *auto_create_table=None*, *partition_keygen=None*, *dynamodb_client=None*)

Bases: *ask_sdk_core.skill_builder.SkillBuilder*

Skill Builder with api client and db adapter coupling to Skill.

Standard Skill Builder is an implementation of *ask_sdk_core.skill_builder.SkillBuilder* with coupling of DynamoDb Persistence Adapter settings and a Default Api Client added to the ask_sdk_core.skill.Skill.

Parameters

- **table_name** (*str*) – Name of the table to be created or used
- **auto_create_table** (*bool*) – Should the adapter try to create the table if it doesn't exist.
- **partition_keygen** (*Callable[[RequestEnvelope], str]*) – Callable function that takes a request envelope and provides a unique partition key value.
- **dynamodb_client** (*boto3.resources.base.ServiceResource*) – Resource to be used, to perform dynamo operations.

add_exception_handler(*exception_handler*)
Register input to the exception handlers list.

Parameters **exception_handler** (*ask_sdk_runtime.dispatch_components.request_components.AbstractExceptionHandler*) – Exception Handler instance to be registered.

Returns None

add_global_request_interceptor(*request_interceptor*)
Register input to the global request interceptors list.

Parameters **request_interceptor** (`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor`) – Request Interceptor instance to be registered.

Returns None

add_global_response_interceptor(*response_interceptor*)
Register input to the global response interceptors list.

Parameters **response_interceptor** (`ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor`) – Response Interceptor instance to be registered.

Returns None

add_request_handler(*request_handler*)
Register input to the request handlers list.

Parameters **request_handler** (`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler`) – Request Handler instance to be registered.

Returns None

create()
Create a skill object using the registered components.

Returns a skill object that can be used for invocation.

Return type Skill

exception_handler(*can_handle_func*)
Decorator that can be used to add exception handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes two parameters and no varargs or kwargs. This is because of the ExceptionHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that processes the exception raised during dispatcher and returns a response object by the skill. The function should follow the signature of the handle function in *`ask_sdk_runtime.dispatch_components.exception_components.AbstractExceptionHandler`* class.

Parameters **can_handle_func** (*`Callable[[Input, Exception], bool]`*) – The function that validates if the exception can be handled.

Returns Wrapper function that can be decorated on a handle function.

global_request_interceptor()
Decorator that can be used to add global request interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input. The function should follow the signature of the process function in *`ask_sdk_runtime.dispatch_components.request_components.AbstractRequestInterceptor`* class.

Returns Wrapper function that can be decorated on a interceptor process function.

global_response_interceptor()
Decorator that can be used to add global response interceptors easily to the builder.

The returned wrapper function can be applied as a decorator on any function that processes the input and the response generated by the request handler. The function should follow the signature of the process function in *`ask_sdk_runtime.dispatch_components.request_components.AbstractResponseInterceptor`* class.

Returns Wrapper function that can be decorated on a interceptor process function.

lambda_handler()
Create a handler function that can be used as handler in AWS Lambda console.

The lambda handler provides a handler function, that acts as an entry point to the AWS Lambda console. Users can set the lambda_handler output to a variable and set the variable as AWS Lambda Handler on the console.

Returns Handler function to tag on AWS Lambda console.

request_handler(*can_handle_func*)
Decorator that can be used to add request handlers easily to the builder.

The can_handle_func has to be a Callable instance, which takes a single parameter and no varargs or kwargs. This is because of the RequestHandler class signature restrictions. The returned wrapper function can be applied as a decorator on any function that returns a response object by the skill. The function should follow the signature of the handle function in `ask_sdk_runtime.dispatch_components.request_components.AbstractRequestHandler` class.

Parameters **can_handle_func** (`Callable[[Input], bool]`) – The function that validates if the request can be handled.

Returns Wrapper function that can be decorated on a handle function.

skill_configuration
Create the skill configuration object using the registered components.

2.10.5 Models

The SDK works on model classes rather than native Alexa JSON requests and responses. These model classes are generated using the Request, Response JSON schemas from the developer docs. The source code for the model classes can be found here.

Subpackages

ask_sdk_model.canfulfill package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.canfulfill.can_fulfill_intent module

class ask_sdk_model.canfulfill.can_fulfill_intent.**CanFulfillIntent**(*can_fulfill=None*, *slots=None*)
Bases: `object`

CanFulfillIntent represents the response to canFulfillIntentRequest includes the details about whether the skill can understand and fulfill the intent request with detected slots.

Parameters

- **can_fulfill** (*(optional)* ask_sdk_model.canfulfill.can_fulfill_intent_values.CanFulfillIntentValues) –
- **slots** (*(optional)* dict(str, ask_sdk_model.canfulfill.can_fulfill_slot.CanFulfillSlot)) – A map that represents skill's detailed response to each detected slot within the intent such as if skill can understand and fulfill the detected slot. This supplements the overall canFulfillIntent response and help Alexa make better ranking and arbitration decisions. The key is the name of the slot. The value is an object of type CanFulfillSlot.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.canfulfill.can_fulfill_intent_request module

class ask_sdk_model.canfulfill.can_fulfill_intent_request.**CanFulfillIntentRequest**(*request_id=I* *timestamp=None*, *locale=None*, *dialog_state=N* *intent=None*)

Bases: *ask_sdk_model.request.Request*

An object that represents a request made to skill to query whether the skill can understand and fulfill the intent request with detected slots, before actually asking the skill to take action. Skill should be aware this is not to actually take action, skill should handle this request without causing side-effect, skill should not modify some state outside its scope or has an observable interaction with its calling functions or the outside world besides returning a value, such as playing sound,turning on/off lights, committing a transaction or a charge.

Parameters

- **request_id** (*(optional)* str) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional)* datetime) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional)* str) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **dialog_state** (*(optional)* ask_sdk_model.dialog_state.DialogState) –
- **intent** (*(optional)* ask_sdk_model.intent.Intent) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.canfulfill.can_fulfill_intent_values module

class ask_sdk_model.canfulfill.can_fulfill_intent_values.**CanFulfillIntentValues**

Bases: `enum.Enum`

Overall if skill can understand and fulfill the intent with detected slots. Respond YES when skill understands all slots, can fulfill all slots, and can fulfill the request in its entirety. Respond NO when skill either cannot understand the intent, cannot understand all the slots, or cannot fulfill all the slots. Respond MAYBE when skill can understand the intent, can partially or fully understand the slots, and can partially or fully fulfill the slots. The only cases where should respond MAYBE is when skill partially understand the request and can potentially complete the request if skill get more data, either through callbacks or through a multi-turn conversation with the user.

Allowed enum values: [YES, NO, MAYBE]

ask_sdk_model.canfulfill.can_fulfill_slot module

class ask_sdk_model.canfulfill.can_fulfill_slot.**CanFulfillSlot**(*can_understand=None*, *can_fulfill=None*)

Bases: `object`

This represents skill's capability to understand and fulfill each detected slot.

Parameters

- **can_understand** (*(optional)* `ask_sdk_model.canfulfill.can_understand_slot_values.CanUnderstandSlotValues`) –
- **can_fulfill** (*(optional)* `ask_sdk_model.canfulfill.can_fulfill_slot_values.CanFulfillSlotValues`) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.canfulfill.can_fulfill_slot_values module

class ask_sdk_model.canfulfill.can_fulfill_slot_values.**CanFulfillSlotValues**

Bases: `enum.Enum`

This field indicates whether skill can fulfill relevant action for the slot, that has been partially or fully understood. The definition of fulfilling the slot is dependent on skill and skill is required to have logic in place to determine whether a slot value can be fulfilled in the context of skill or not. Return YES if Skill can certainly fulfill the relevant action for this slot value. Return NO if skill cannot fulfill the relevant action for this slot value. For specific recommendations to set the value refer to the developer docs for more details.

Allowed enum values: [YES, NO]

ask_sdk_model.canfulfill.can_understand_slot_values module

class ask_sdk_model.canfulfill.can_understand_slot_values.**CanUnderstandSlotValues**

Bases: `enum.Enum`

This field indicates whether skill has understood the slot value. In most typical cases, skills will do some form of entity resolution by looking up a catalog or list to determine whether they recognize the slot or not. Return YES if skill have a perfect match or high confidence match (for eg. synonyms) with catalog or list maintained by skill. Return NO if skill cannot understand or recognize the slot value. Return MAYBE if skill have partial confidence or partial match. This will be true when the slot value doesn't exist as is, in the catalog, but a variation or a fuzzy match may exist. For specific recommendations to set the value refer to the developer docs for more details.

Allowed enum values: [YES, NO, MAYBE]

ask_sdk_model.dialog package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.dialog.confirm_intent_directive module

class `ask_sdk_model.dialog.confirm_intent_directive.`**`ConfirmIntentDirective`**(*updated_intent=None*)

Bases: *`ask_sdk_model.directive.Directive`*

Parameters **`updated_intent`** (*`(optional)`* `ask_sdk_model.intent.Intent`) –

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.dialog.confirm_slot_directive module

class `ask_sdk_model.dialog.confirm_slot_directive.`**`ConfirmSlotDirective`**(*updated_intent=None, slot_to_confirm=None*)

Bases: *`ask_sdk_model.directive.Directive`*

Parameters

- **`updated_intent`** (*`(optional)`* `ask_sdk_model.intent.Intent`) –
- **`slot_to_confirm`** (*`(optional)`* *`str`*) –

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.dialog.delegate_directive module

class ask_sdk_model.dialog.delegate_directive.**DelegateDirective**(*updated_intent=None*)

Bases: *ask_sdk_model.directive.Directive*

Parameters **updated_intent** (*(optional)* ask_sdk_model.intent.Intent) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.dialog.elicit_slot_directive module

class ask_sdk_model.dialog.elicit_slot_directive.**ElicitSlotDirective**(*updated_intent=None*, *slot_to_elicit=None*)

Bases: *ask_sdk_model.directive.Directive*

Parameters

- **updated_intent** (*(optional)* ask_sdk_model.intent.Intent) –
- **slot_to_elicit** (*(optional)* *str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.events package

Subpackages

ask_sdk_model.events.skillevents package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.events.skillevents.account_linked_body module

class ask_sdk_model.events.skillevents.account_linked_body.**AccountLinkedBody**(*access_token=None*

Bases: `object`

Parameters **access_token** (*(optional)* *str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.events.skillevents.account_linked_request module

class ask_sdk_model.events.skillevents.account_linked_request.**AccountLinkedRequest**(*request_id= timestamp=Non locale=None body=Non event_crea event_publ*

Bases: *ask_sdk_model.request.Request*

This event indicates that a customer has linked an account in a third-party application with the Alexa app. This event is useful for an application that support out-of-session (non-voice) user interactions so that this application can be notified when the internal customer can be associated with the Alexa customer. This event is required for many applications that synchronize customer Alexa lists with application lists. During the account linking process, the Alexa app directs the user to the skill website where the customer logs in. When the customer logs in, the skill then provides an access token and a consent token to Alexa. The event includes the same access token and consent token.

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional) ask_sdk_model.events.skillevents.account_linked_body.AccountLinkedBody*) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.events.skillevents.permission module

class ask_sdk_model.events.skillevents.permission.**Permission**(*scope=None*)
Bases: object

Parameters **scope** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.events.skillevents.permission_accepted_request module

class ask_sdk_model.events.skillevents.permission_accepted_request.**PermissionAcceptedReques**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.events.skillevents.permission_body.PermissionBody) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.events.skillevents.permission_body module

class ask_sdk_model.events.skillevents.permission_body.**PermissionBody**(*accepted_permissions=None*)
Bases: object

Parameters accepted_permissions (*(optional) list[*ask_sdk_model.events.skillevents.permission.Permission*]*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.events.skillevents.permission_changed_request module

class ask_sdk_model.events.skillevents.permission_changed_request.**PermissionChangedRequest**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.events.skillevents.permission_body.PermissionBody) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.events.skillevents.skill_disabled_request module

class ask_sdk_model.events.skillevents.skill_disabled_request.**SkillDisabledRequest**(*request_id= timestamp=Non locale=None event_crea event_publ*

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.events.skillevents.skill_enabled_request module

class ask_sdk_model.events.skillevents.skill_enabled_request.**SkillEnabledRequest**(*request_id=N timestamp=None, locale=None, event_creation event_publish*

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces package

Subpackages

ask_sdk_model.interfaces.alexa package

Subpackages

ask_sdk_model.interfaces.alexa.presentation package

Subpackages

ask_sdk_model.interfaces.alexa.presentation.apl package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

---

ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface module

class `ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface.`**`Ale`**

Bases: `object`

Parameters runtime (*(optional)* `ask_sdk_model.interfaces.alexa.presentation.apl.runtime.Runtime`) –

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.align module

class `ask_sdk_model.interfaces.alexa.presentation.apl.align.`**`Align`**

Bases: `enum.Enum`

The alignment of the item after scrolling. Defaults to visible.

Allowed enum values: [center, first, last, visible]

ask_sdk_model.interfaces.alexa.presentation.apl.auto_page_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.auto_page_command.**AutoPageCommand**(*dela description when component count duration*

Bases: *ask_sdk_model.interfaces.alexa.presentation.apl.command.Command*

Automatically progress through a series of pages displayed in a Pager component. The AutoPage command finishes after the last page has been displayed for the requested time period.

Parameters

- **delay** (*(optional) int*) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- **description** (*(optional) str*) – A user-provided description of this command.
- **when** (*(optional) bool*) – If false, the execution of the command is skipped. Defaults to true.
- **component_id** (*(optional) str*) – The id of the Pager component.
- **count** (*(optional) int*) – Number of pages to display. Defaults to all of them.
- **duration** (*(optional) int*) – Time to wait between pages (in milliseconds). Defaults to 0.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.command module

class ask_sdk_model.interfaces.alexa.presentation.apl.command.**Command**(*object_type=None*, *delay=None*, *description=None*, *when=None*)

Bases: object

A message that can change the visual or audio presentation of the content on the screen.

Parameters

- **object_type** (*(optional) str*) – Defines the command type and dictates which properties must/can be included.

- **delay** (*(optional) int*) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- **description** (*(optional) str*) – A user-provided description of this command.
- **when** (*(optional) bool*) – If false, the execution of the command is skipped. Defaults to true.

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

SetPage: *ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command.SetPageCommand*,

SpeakItem: *ask_sdk_model.interfaces.alexa.presentation.apl.speak_item_command.SpeakItemCommand*,

AutoPage: *ask_sdk_model.interfaces.alexa.presentation.apl.auto_page_command.AutoPageCommand*

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.execute_commands_directive module

class ask_sdk_model.interfaces.alexa.presentation.apl.execute_commands_directive.**ExecuteCo**

Bases: *ask_sdk_model.directive.Directive*

Alexa.Presentation.APL.ExecuteCommands directive used to send APL commands to a device.

Parameters

- **commands** (*(optional) list[ask_sdk_model.interfaces.alexa.presentation.apl.command.Command]*) – List of Command instances
- **token** (*(optional) str*) – A skill defined token, unique for each presentation. Must match the token provided by the skill in the RenderDocument directive used to render the original APL document.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode module

class ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode.**HighlightMode**

Bases: enum.Enum

How highlighting is applied: on a line-by-line basis, or to the entire block. Defaults to block.

Allowed enum values: [block, line]

ask_sdk_model.interfaces.alexa.presentation.apl.position module

class ask_sdk_model.interfaces.alexa.presentation.apl.position.**Position**

Bases: enum.Enum

Whether the value is a relative or absolute offset. Defaults to absolute.

Allowed enum values: [absolute, relative]

ask_sdk_model.interfaces.alexa.presentation.apl.render_document_directive module

class ask_sdk_model.interfaces.alexa.presentation.apl.render_document_directive.**RenderDocu**

Bases: *ask_sdk_model.directive.Directive*

Parameters

- **token** (*(optional) str*) – A unique identifier for the presentation.
- **document** (*(optional) dict(str, object)*) – The APL document that the devices need to render a presentation.
- **datasources** (*(optional) dict(str, object)*) – Data sources to bind to the document when rendering.
- **packages** (*(optional) list[object]*) – A list of packages including layouts, styles, and images etc.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.runtime module

class ask_sdk_model.interfaces.alexa.presentation.apl.runtime.**Runtime**(*max_version=None*)

Bases: object

Contains the runtime information for the interface.

Parameters **max_version** (*(optional) str*) – Maximum APL version supported by the runtime.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.set_page_command.**SetPageCommand**(*delay= description=N when= component_i position=N value=*

Bases: *ask_sdk_model.interfaces.alexa.presentation.apl.command.Command*

Change the page displayed in a Pager component. The SetPage command finishes when the item is fully in view.

Parameters

- **delay** (*(optional) int*) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- **description** (*(optional) str*) – A user-provided description of this command.
- **when** (*(optional) bool*) – If false, the execution of the command is skipped. Defaults to true.
- **component_id** (*(optional) str*) – The id of the Pager component.
- **position** (*(optional) ask_sdk_model.interfaces.alexa.presentation.apl.position.Position*) –
- **value** (*(optional) int*) – The distance to move. May be an absolute value or a relative value.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.speak_item_command module

class ask_sdk_model.interfaces.alexa.presentation.apl.speak_item_command.**SpeakItemCommand**(*d*
d
s
t
w
a
c
c
p
r
h
l
r
i
r

Bases: *ask_sdk_model.interfaces.alexa.presentation.apl.command.Command*

Reads the contents of a single item on the screen. By default the item will be scrolled into view if it is not currently visible.

Parameters

- **delay** (*(optional) int*) – The delay in milliseconds before this command starts executing; must be non-negative. Defaults to 0.
- **description** (*(optional) str*) – A user-provided description of this command.
- **when** (*(optional) bool*) – If false, the execution of the command is skipped. Defaults to true.
- **align** (*(optional)* ask_sdk_model.interfaces.alexa.presentation.apl.align.Align) –
- **component_id** (*(optional) str*) – The id of the component to speak.
- **highlight_mode** (*(optional)* ask_sdk_model.interfaces.alexa.presentation.apl.highlight_mode.HighlightMode) –
- **minimum_dwell_time** (*(optional) int*) – The minimum number of milliseconds that an item should be highlighted for. Defaults to 0.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.alexa.presentation.apl.user_event module

class ask_sdk_model.interfaces.alexa.presentation.apl.user_event.**UserEvent**(*request_id=None*, *timestamp=None*, *locale=None*, *token=None*, *arguments=None*, *source=None*, *components=None*)

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **token** (*(optional) str*) – A unique token for the active presentation.
- **arguments** (*(optional) list[object]*) – The array of argument data to pass to Alexa.
- **source** (*(optional) object*) – Meta-information about what caused the event to be generated.
- **components** (*(optional) object*) – Components associated with the request.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

Module contents

Module contents

Module contents

ask_sdk_model.interfaces.amazonpay package

Subpackages

ask_sdk_model.interfaces.amazonpay.model package

Subpackages

ask_sdk_model.interfaces.amazonpay.model.v1 package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.amazonpay.model.v1.authorization_details module

class ask_sdk_model.interfaces.amazonpay.model.v1.authorization_details.**AuthorizationDetai**

Bases: `object`

This object encapsulates details about an Authorization object including the status, amount captured and fee charged.

Parameters

- **amazon_authorization_id** (*(optional)* *str*) – This is AmazonPay generated identifier for this authorization transaction.
- **authorization_reference_id** (*(optional)* *str*) – This is 3P seller's identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.
- **seller_authorization_note** (*(optional)* *str*) – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.
- **authorization_amount** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.price.Price) –
- **captured_amount** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.price.Price) –

- **authorization_fee** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.price.Price) –
- **id_list** (*(optional)* *list[str]*) – list of AmazonCaptureId identifiers that have been requested on this Authorization object.
- **creation_timestamp** (*(optional)* *datetime*) – This is the time at which the authorization was created.
- **expiration_timestamp** (*(optional)* *datetime*) – This is the time at which the authorization expires.
- **authorization_status** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status.AuthorizationStatus) –
- **soft_decline** (*(optional)* *bool*) – This indicates whether an authorization resulted in a soft decline.
- **capture_now** (*(optional)* *bool*) – This indicates whether a direct capture against the payment contract was specified.
- **soft_descriptor** (*(optional)* *str*) – This is the description to be shown on the buyer's payment instrument statement if AuthorizeAndCapture was chosen.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status module

class ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status.**AuthorizationStatus**

Bases: object

Indicates the current status of an Authorization object, a Capture object, or a Refund object.

Parameters

- **state** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.state.State) –
- **reason_code** (*(optional)* *str*) – The reason that the Authorization object, Capture object, or Refund object is in the current state. For more information, see - https://pay.amazon.com/us/developer/documentation/apireference/201752950
- **reason_description** (*(optional)* *str*) – Reason desciption corresponding to the reason code
- **last_update_timestamp** (*(optional)* *datetime*) – A timestamp that indicates the time when the authorization, capture, or refund state was last updated. In ISO 8601 format

to_dict()
Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.authorize_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.authorize_attributes.**AuthorizeAttributes**

Bases: `object`

This is an object to set the attributes specified in the AuthorizeAttributes table. See the "AuthorizationDetails" section of the Amazon Pay API reference guide for details about this object.

Parameters

- **authorization_reference_id** (*(optional) str*) – This is 3P seller's identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.
- **authorization_amount** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.v1.price.Price`) –
- **transaction_timeout** (*(optional) int*) – The maximum number of minutes allocated for the Authorize operation call to be processed. After this the authorization is automatically declined and you cannot capture funds against the authorization. The default value for Alexa transactions is 0. In order to speed up checkout time for voice users we recommend to not change this value.
- **seller_authorization_note** (*(optional) str*) – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.
- **soft_descriptor** (*(optional) str*) – The description to be shown on the user's payment instrument statement if AuthorizeAndCapture is chosen. Format of soft descriptor sent to the payment processor is &quot;AMZ* &lt;soft descriptor specified here&gt;&quot;. Default is &quot;AMZ*&lt;SELLER_NAME&gt; amzn.com/ pmts WA&quot;. Maximum length can be 16 characters.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes.**BillingAgree**

Bases: `object`

The merchant can choose to set the attributes specified in the BillingAgreementAttributes.

Parameters

- **platform_id** (*(optional) str*) – Represents the SellerId of the Solution Provider that developed the eCommerce platform. This value is only used by Solution Providers, for whom it is required. It should not be provided by merchants creating their own custom integration. Do not specify the SellerId of the merchant for this request parameter. If you are a merchant, do not enter a PlatformId.
- **seller_note** (*(optional) str*) – Represents a description of the billing agreement that is displayed in emails to the buyer.
- **seller_billing_agreement_attributes** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes.SellerBillingAgreementAttributes) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details module

class ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details.**BillingAgreemen**

Bases: object

The result attributes from successful SetupAmazonPay call.

Parameters

- **billing_agreement_id** (*(optional) str*) – Billing agreement id which can be used for one time and recurring purchases
- **creation_timestamp** (*(optional) datetime*) – Time at which billing agreement details created.
- **destination** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.destination.Destination) –
- **checkout_language** (*(optional) str*) – Merchant's preferred language of checkout.
- **release_environment** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.release_environment.ReleaseEnvironment) –

- **billing_agreement_status** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status.BillingAgreementStatus) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status module

class ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status.**BillingAgreement**

Bases: enum.Enum

Indicates the current status of the billing agreement. For more information about the State and ReasonCode response elements, see Billing agreement states and reason codes - https://pay.amazon.com/us/developer/documentation/apireference/201752870

Allowed enum values: [CANCELED, CLOSED, DRAFT, OPEN, SUSPENDED]

ask_sdk_model.interfaces.amazonpay.model.v1.destination module

class ask_sdk_model.interfaces.amazonpay.model.v1.destination.**Destination**(*name=None, company_name=None, address_line1=None, address_line2=None, address_line3=None, city=None, district_or_county=None, state_or_region=None, postal_code=None, country_code=None, phone=None*)

Bases: object

The default shipping address of the buyer. Returned if needAmazonShippingAddress is set to true.

Parameters

- **name** (*(optional) str*) – The name or business name
- **company_name** (*(optional) str*) – The company name
- **address_line1** (*(optional) str*) – The first line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.
- **address_line2** (*(optional) str*) – The second line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.
- **address_line3** (*(optional) str*) – The third line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.

- **city** (*(optional) str*) – The city
- **district_or_county** (*(optional) str*) – The district or County
- **state_or_region** (*(optional) str*) – The state or region. This element is free text and can be either a 2-character code, fully spelled out, or abbreviated. Required. Note :- This response element is returned only in the U.S.
- **postal_code** (*(optional) str*) – The postal code.
- **country_code** (*(optional) str*) – The country code, in ISO 3166 format
- **phone** (*(optional) str*) – The phone number

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.payment_action module

class ask_sdk_model.interfaces.amazonpay.model.v1.payment_action.**PaymentAction**
Bases: enum.Enum

- This is used to specify applicable payment action. * Authorize – you want to confirm the order and authorize a certain amount, but you do not want to capture at this time. * AuthorizeAndCapture – you want to confirm the order, authorize for the given amount, and capture the funds.

Allowed enum values: [Authorize, AuthorizeAndCapture]

ask_sdk_model.interfaces.amazonpay.model.v1.price module

class ask_sdk_model.interfaces.amazonpay.model.v1.price.**Price**(*amount=None, currency_code=None*)
Bases: object

This object specifies amount and currency authorized/captured.

Parameters

- **amount** (*(optional) str*) – Amount authorized/captured.
- **currency_code** (*(optional) str*) – Currency code for the amount.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes.**ProviderAttributes**(*pr pr*
Bases: object

This is required only for Ecommerce provider (Solution provider) use cases.

Parameters

- **provider_id** (*(optional) str*) – Solution provider ID.
- **provider_credit_list** (*(optional) list[ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit.ProviderCredit]*) – List of provider credit.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit module

class ask_sdk_model.interfaces.amazonpay.model.v1.provider_credit.**ProviderCredit**(*provider_id=I* *credit=None*)

Bases: object

Parameters

- **provider_id** (*(optional) str*) – This is required only for Ecommerce provider (Solution provider) use cases.
- **credit** (*(optional) ask_sdk_model.interfaces.amazonpay.model.v1.price.Price*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.release_environment module

class ask_sdk_model.interfaces.amazonpay.model.v1.release_environment.**ReleaseEnvironment**

Bases: enum.Enum

Indicates if the order is for a Live (Production) or Sandbox environment.

Allowed enum values: [LIVE, SANDBOX]

ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.seller_billing_agreement_attributes.**Selle**

Bases: object

Provides more context about the billing agreement that is represented by this Billing Agreement object.

Parameters

- **seller_billing_agreement_id** (*(optional) str*) – The merchant-specified identifier of this billing agreement. At least one request parameter must be specified. Amazon recommends that you use only the following characters:- lowercase a-z, uppercase A-Z, numbers 0-9, dash (-), underscore (_).

- **store_name** (*(optional) str*) – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.
- **custom_information** (*(optional) str*) – Any additional information that you wish to include with this billing agreement. At least one request parameter must be specified.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes module

class ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes.**SellerOrderAttril**

Bases: object

This object includes elements shown to buyers in emails and in their transaction history. See the "SellerOrderAttributes" section of the Amazon Pay API reference guide for details about this object.

Parameters

- **seller_order_id** (*(optional) str*) – The merchant-specified identifier of this order. This is shown to the buyer in their emails and transaction history on the Amazon Pay website.
- **store_name** (*(optional) str*) – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.
- **custom_information** (*(optional) str*) – Any additional information that you want to include with this order reference.
- **seller_note** (*(optional) str*) – This represents a description of the order that is displayed in emails to the buyer.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.v1.state module

class ask_sdk_model.interfaces.amazonpay.model.v1.state.**State**

Bases: enum.Enum

Indicates the state that the Authorization object, Capture object, or Refund object is in. For more information see - https://pay.amazon.com/us/developer/documentation/apireference/201752950

Allowed enum values: [Pending, Open, Declined, Closed, Completed]

ask_sdk_model.interfaces.amazonpay.model.request package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes module

class `ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes.`**`AuthorizeAttrib`**

Bases: *`ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity`*

This is an object to set the attributes specified in the AuthorizeAttributes table. See the “AuthorizationDetails” section of the Amazon Pay API reference guide for details about this object.

Parameters

- **`authorization_reference_id`** (*(optional)* *str*) – This is 3P seller’s identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.
- **`authorization_amount`** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.request.price.Price`) –
- **`transaction_timeout`** (*(optional)* *int*) – The maximum number of minutes allocated for the Authorize operation call to be processed. After this the authorization is automatically declined and you cannot capture funds against the authorization. The default value for Alexa transactions is 0. In order to speed up checkout time for voice users we recommend to not change this value.
- **`seller_authorization_note`** (*(optional)* *str*) – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.
- **`soft_descriptor`** (*(optional)* *str*) – The description to be shown on the user’s payment instrument statement if AuthorizeAndCapture is chosen. Format of soft descriptor sent to the payment processor is &quot;AMZ* &lt;soft descriptor specified here&gt;&quot;. Default is &quot;AMZ*&lt;SELLER_NAME&gt; amzn.com/ pmts WA&quot;. Maximum length can be 16 characters.

- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity module

class ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.**BaseAmazonPay**

Bases: object

Parameters

- **object_type** (*(optional) str*) –
- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets @type variable.

AuthorizeAttributes: *ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes.AuthorizeAttributes*,

SellerBillingAgreementAttributes: *ask_sdk_model.interfaces.amazonpay.model.request.seller_billing_agreement_attributes.SellerBillingAgreementAttributes*,

SetupAmazonPayRequest: *ask_sdk_model.interfaces.amazonpay.request.setup_amazon_pay_request.SetupAmazonPayRequest*,

ProviderCredit: *ask_sdk_model.interfaces.amazonpay.model.request.provider_credit.ProviderCredit*,

Price: *ask_sdk_model.interfaces.amazonpay.model.request.price.Price*,

ChargeAmazonPayRequest: *ask_sdk_model.interfaces.amazonpay.request.charge_amazon_pay_request.ChargeAmazonPayRequest*,

BillingAgreementAttributes: *ask_sdk_model.interfaces.amazonpay.model.request.billing_agreement_attributes.BillingAgreementAttributes*,

SellerOrderAttributes: *ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes.SellerOrderAttributes*,

ProviderAttributes: *ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes.ProviderAttributes*

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.request.billing_agreement_attributes module

class ask_sdk_model.interfaces.amazonpay.model.request.billing_agreement_attributes.**Billing**

Bases: *ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity*

The merchant can choose to set the attributes specified in the BillingAgreementAttributes.

Parameters

- **platform_id** (*(optional) str*) – Represents the SellerId of the Solution Provider that developed the eCommerce platform. This value is only used by Solution Providers, for whom it is required. It should not be provided by merchants creating their own custom integration. Do not specify the SellerId of the merchant for this request parameter. If you are a merchant, do not enter a PlatformId.
- **seller_note** (*(optional) str*) – Represents a description of the billing agreement that is displayed in emails to the buyer.
- **seller_billing_agreement_attributes** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.request.seller_billing_agreement_attributes.SellerBillingAgreementAttributes) –
- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.request.payment_action module

class ask_sdk_model.interfaces.amazonpay.model.request.payment_action.**PaymentAction**
Bases: enum.Enum

- This is used to specify applicable payment action. * Authorize – you want to confirm the order and authorize a certain amount, but you do not want to capture at this time. * AuthorizeAndCapture – you want to confirm the order, authorize for the given amount, and capture the funds.

Allowed enum values: [Authorize, AuthorizeAndCapture]

ask_sdk_model.interfaces.amazonpay.model.request.price module

class ask_sdk_model.interfaces.amazonpay.model.request.price.**Price**(*amount=None, currency_code=None, version=None*)

Bases: *ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity*

This request object specifies amount and currency authorized/captured.

Parameters

- **amount** (*(optional) str*) – Amount authorized/captured.
- **currency_code** (*(optional) str*) – Currency code for the amount.
- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes module

class ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes.**ProviderAttribut**

Bases: *ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity*

This is required only for Ecommerce provider (Solution provider) use cases.

Parameters

- **provider_id** (*(optional) str*) – Solution provider ID.
- **provider_credit_list** (*(optional) list[ask_sdk_model.interfaces.amazonpay.model.request.provider_credit.ProviderCredit]*) – List of provider credit.
- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.request.provider_credit module

class ask_sdk_model.interfaces.amazonpay.model.request.provider_credit.**ProviderCredit**(*provid credit= version=*

Bases: *ask_sdk_model.interfaces.amazonpay.model.request.*

base_amazon_pay_entity.BaseAmazonPayEntity

Parameters

- **provider_id** (*(optional) str*) – This is required only for Ecommerce provider (Solution provider) use cases.
- **credit** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.request.price.Price) –
- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.request.seller_billing_agreement_attributes module

class ask_sdk_model.interfaces.amazonpay.model.request.seller_billing_agreement_attributes

Bases: *ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity*

Provides more context about the billing agreement that is represented by this Billing Agreement object.

Parameters

- **seller_billing_agreement_id** (*(optional) str*) – The merchant-specified identifier of this billing agreement. At least one request parameter must be specified. Amazon recommends that you use only the following characters:- lowercase a-z, uppercase A-Z, numbers 0-9, dash (-), underscore (_).
- **store_name** (*(optional) str*) – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.
- **custom_information** (*(optional) str*) – Any additional information that you wish to include with this billing agreement. At least one request parameter must be specified.
- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes module

class ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes.**SellerOrderA**

Bases: *ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity*

This object includes elements shown to buyers in emails and in their transaction history. See the “SellerOrderAttributes” section of the Amazon Pay API reference guide for details about this object.

Parameters

- **seller_order_id** (*(optional) str*) – The merchant-specified identifier of this order. This is shown to the buyer in their emails and transaction history on the Amazon Pay website.
- **store_name** (*(optional) str*) – The identifier of the store from which the order was placed. This overrides the default value in Seller Central under Settings &gt; Account Settings. It is displayed to the buyer in their emails and transaction history on the Amazon Payments website.
- **custom_information** (*(optional) str*) – Any additional information that you want to include with this order reference.
- **seller_note** (*(optional) str*) – This represents a description of the order that is displayed in emails to the buyer.
- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.response package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.amazonpay.model.response.authorization_details module

class ask_sdk_model.interfaces.amazonpay.model.response.authorization_details.**Authorizatio**

Bases: *ask_sdk_model.interfaces.amazonpay.model.v1.authorization_details.AuthorizationDetails*

This object encapsulates details about an Authorization object including the status, amount captured and fee charged.

Parameters

- **amazon_authorization_id** (*(optional) str*) – This is AmazonPay generated identifier for this authorization transaction.
- **authorization_reference_id** (*(optional) str*) – This is 3P seller's identifier for this authorization transaction. This identifier must be unique for all of your authorization transactions.
- **seller_authorization_note** (*(optional) str*) – A description for the transaction that is included in emails to the user. Appears only when AuthorizeAndCapture is chosen.
- **authorization_amount** (*(optional)* ask_sdk_model.interfaces.

amazonpay.model.response.price.Price) –

- **captured_amount** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.price.Price) –
- **authorization_fee** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.price.Price) –
- **id_list** (*(optional) list[str]*) – list of AmazonCaptureId identifiers that have been requested on this Authorization object.
- **creation_timestamp** (*(optional) datetime*) – This is the time at which the authorization was created.
- **expiration_timestamp** (*(optional) datetime*) – This is the time at which the authorization expires.
- **authorization_status** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.authorization_status.AuthorizationStatus) –
- **soft_decline** (*(optional) bool*) – This indicates whether an authorization resulted in a soft decline.
- **capture_now** (*(optional) bool*) – This indicates whether a direct capture against the payment contract was specified.
- **soft_descriptor** (*(optional) str*) – This is the description to be shown on the buyer's payment instrument statement if AuthorizeAndCapture was chosen.
- **authorization_billing_address** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.destination.Destination) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.response.authorization_status module

class ask_sdk_model.interfaces.amazonpay.model.response.authorization_status.**Authorization**

Bases: *ask_sdk_model.interfaces.amazonpay.model.v1.authorization_status.AuthorizationStatus*

Indicates the current status of an Authorization object, a Capture object, or a Refund object.

Parameters

- **state** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.state.State) –
- **reason_code** (*(optional) str*) – The reason that the Authorization object, Capture object, or Refund object is in the current state. For more information, see - https://pay.amazon.com/us/developer/documentation/apireference/201752950

- **reason_description** (*(optional) str*) – Reason desciption corresponding to the reason code
- **last_update_timestamp** (*(optional) datetime*) – A timestamp that indicates the time when the authorization, capture, or refund state was last updated. In ISO 8601 format

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.response.billing_agreement_details module

class ask_sdk_model.interfaces.amazonpay.model.response.billing_agreement_details.**BillingAg**

Bases: *ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details.BillingAgreementDetails*

The result attributes from successful SetupAmazonPay call.

Parameters

- **billing_agreement_id** (*(optional) str*) – Billing agreement id which can be used for one time and recurring purchases
- **creation_timestamp** (*(optional) datetime*) – Time at which billing agreement details created.
- **destination** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.destination.Destination) –
- **checkout_language** (*(optional) str*) – Merchant's preferred language of checkout.
- **release_environment** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.release_environment.ReleaseEnvironment) –
- **billing_agreement_status** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_status.BillingAgreementStatus) –
- **billing_address** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.destination.Destination) –

to_dict()
Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.response.destination module

class ask_sdk_model.interfaces.amazonpay.model.response.destination.**Destination**(*name=None, company_name=No address_line1=No address_line2=No address_line3=No city=None, district_or_county state_or_region postal_code=N country_code=None phone=None*)

Bases: *ask_sdk_model.interfaces.amazonpay.model.v1.destination.Destination*

The default shipping address of the buyer. Returned if needAmazonShippingAddress is set to true.

Parameters

- **name** (*(optional) str*) – The name or business name
- **company_name** (*(optional) str*) – The company name
- **address_line1** (*(optional) str*) – The first line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.
- **address_line2** (*(optional) str*) – The second line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.
- **address_line3** (*(optional) str*) – The third line of the address. At least one AddressLine (AddressLine1, AddressLine2, or AddressLine3) is required.
- **city** (*(optional) str*) – The city
- **district_or_county** (*(optional) str*) – The district or County
- **state_or_region** (*(optional) str*) – The state or region. This element is free text and can be either a 2-character code, fully spelled out, or abbreviated. Required. Note :- This response element is returned only in the U.S.
- **postal_code** (*(optional) str*) – The postal code.
- **country_code** (*(optional) str*) – The country code, in ISO 3166 format
- **phone** (*(optional) str*) – The phone number

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.response.price module

class ask_sdk_model.interfaces.amazonpay.model.response.price.**Price**(*amount=None*, *currency_code=None*)

Bases: *ask_sdk_model.interfaces.amazonpay.model.v1.price.Price*

This response object specifies amount and currency authorized/captured.

Parameters

- **amount** (*(optional) str*) – Amount authorized/captured.
- **currency_code** (*(optional) str*) – Currency code for the amount.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.model.response.release_environment module

class ask_sdk_model.interfaces.amazonpay.model.response.release_environment.**ReleaseEnvironm**

Bases: enum.Enum

Allowed enum values: [LIVE, SANDBOX]

ask_sdk_model.interfaces.amazonpay.model.response.state module

class ask_sdk_model.interfaces.amazonpay.model.response.state.**State**

Bases: enum.Enum

Indicates the state that the Authorization object is in. For more information see “Authorization states and reason codes” under “States and reason codes” section in Amazon Pay API Reference Guide.

Allowed enum values: [Pending, Open, Declined, Closed]

ask_sdk_model.interfaces.amazonpay.v1 package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.amazonpay.v1.amazon_pay_error_response module

class ask_sdk_model.interfaces.amazonpay.v1.amazon_pay_error_response.**AmazonPayErrorRespons**

Bases: `object`

Error response for SetupAmazonPay and ChargeAmazonPay calls.

Parameters

- **error_code** (*(optional)* `str`) – Error code indicating the succinct cause of error
- **error_message** (*(optional)* `str`) – Description of the error.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay module

class ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay.**ChargeAmazonPay**(*consent_token=No seller_id=None, billing_agreement payment_action=None authorize_attributes=No seller_order_attrib provider_attribute*

Bases: `object`

Charge Amazon Pay Request Object

Parameters

- **consent_token** (*(optional)* `str`) – Authorization token that contains the permissions consented to by the user.
- **seller_id** (*(optional)* `str`) – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.
- **billing_agreement_id** (*(optional)* `str`) – The payment contract i.e. billing agreement created for the user.
- **payment_action** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.v1.payment_action.PaymentAction`) –
- **authorize_attributes** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.v1.authorize_attributes.AuthorizeAttributes`) –
- **seller_order_attributes** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.v1.seller_order_attributes.SellerOrderAttributes`) –

- **provider_attributes** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.provider_attributes.ProviderAttributes) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay_result module

class ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay_result.**ChargeAmazonPayResult**

Bases: object

Charge Amazon Pay Result Object. It is sent as part of the reponse to ChargeAmazonPay request.

Parameters

- **amazon_order_reference_id** (*(optional)* str) – The order reference identifier.
- **authorization_details** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.authorization_details.AuthorizationDetails) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay module

class ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay.**SetupAmazonPay**(*consent_token=None, seller_id=None, country_of_establishment= ledger_currency=Nor checkout_language=None, billing_agreement_at need_amazon_shippi sandbox_mode=False, sandbox_customer_email_*

Bases: object

Setup Amazon Pay Request Object

Parameters

- **consent_token** (*(optional) str*) – Authorization token that contains the permissions consented to by the user.
- **seller_id** (*(optional) str*) – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.
- **country_of_establishment** (*(optional) str*) – The country in which the merchant has registered, as an Amazon Payments legal entity.
- **ledger_currency** (*(optional) str*) – The currency of the merchant's ledger account.
- **checkout_language** (*(optional) str*) – The merchant's preferred language for checkout.
- **billing_agreement_attributes** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_attributes.BillingAgreementAttributes) –
- **need_amazon_shipping_address** (*bool*) – To receive the default user shipping address in the response, set this parameter to true. Not required if a user shipping address is not required.
- **sandbox_mode** (*bool*) – To test in Sandbox mode, set this parameter to true.
- **sandbox_customer_email_id** (*(optional) str*) – Use this parameter to create a Sandbox payment object. In order to use this parameter, you first create a Sandbox user account in Seller Central. Then, pass the email address associated with that Sandbox user account.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay_result module

class ask_sdk_model.interfaces.amazonpay.v1.setup_amazon_pay_result.**SetupAmazonPayResult**(*bi*

Bases: object

Setup Amazon Pay Result Object. It is sent as part of the reponse to SetupAmazonPay request.

Parameters billing_agreement_details (*(optional)* ask_sdk_model.interfaces.amazonpay.model.v1.billing_agreement_details.BillingAgreementDetails) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.request package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.amazonpay.request.charge_amazon_pay_request module

class `ask_sdk_model.interfaces.amazonpay.request.charge_amazon_pay_request.`**`ChargeAmazonPayI`**

Bases: *`ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity`*

Charge Amazon Pay Request Object.

Parameters

- **`version`** (*(optional)* `str`) – Version of the Amazon Pay Entity. Can be 2 or greater.
- **`seller_id`** (*(optional)* `str`) – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.
- **`billing_agreement_id`** (*(optional)* `str`) – The payment contract i.e. billing agreement created for the user.
- **`payment_action`** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.request.payment_action.PaymentAction`) –
- **`authorize_attributes`** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.request.authorize_attributes.AuthorizeAttributes`) –
- **`seller_order_attributes`** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.request.seller_order_attributes.SellerOrderAttributes`) –
- **`provider_attributes`** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.request.provider_attributes.ProviderAttributes`) –

`to_dict``()`

Returns the model properties as a dict

`to_str``()`

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.request.setup_amazon_pay_request module

class ask_sdk_model.interfaces.amazonpay.request.setup_amazon_pay_request.**SetupAmazonPayRe**

Bases: *ask_sdk_model.interfaces.amazonpay.model.request.base_amazon_pay_entity.BaseAmazonPayEntity*

Setup Amazon Pay Request Object.

Parameters

- **version** (*(optional) str*) – Version of the Amazon Pay Entity. Can be 2 or greater.
- **seller_id** (*(optional) str*) – The seller ID (also known as merchant ID). If you are an Ecommerce Provider (Solution Provider), please specify the ID of the merchant, not your provider ID.
- **country_of_establishment** (*(optional) str*) – The country in which the merchant has registered, as an Amazon Payments legal entity.
- **ledger_currency** (*(optional) str*) – The currency of the merchant's ledger account.
- **checkout_language** (*(optional) str*) – The merchant's preferred language for checkout.
- **billing_agreement_attributes** (*(optional)* ask_sdk_model.interfaces.amazonpay.model.request.billing_agreement_attributes.BillingAgreementAttributes) –
- **need_amazon_shipping_address** (*bool*) – To receive the default user shipping address in the response, set this parameter to true. Not required if a user shipping address is not required.
- **sandbox_mode** (*bool*) – To test in Sandbox mode, set this parameter to true.
- **sandbox_customer_email_id** (*(optional) str*) – Use this parameter to create a Sandbox payment object. In order to use this parameter, you first create a Sandbox user account in Seller Central. Then, pass the email address associated with that Sandbox user account.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.response package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.amazonpay.response.amazon_pay_error_response module

class `ask_sdk_model.interfaces.amazonpay.response.amazon_pay_error_response.`**`AmazonPayErrorR`**

Bases: *`ask_sdk_model.interfaces.amazonpay.v1.amazon_pay_error_response.AmazonPayErrorResponse`*

Error response for SetupAmazonPay and ChargeAmazonPay calls.

Parameters

- **`error_code`** (*(optional)* *str*) – Error code indicating the succinct cause of error
- **`error_message`** (*(optional)* *str*) – Description of the error.

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.response.charge_amazon_pay_result module

class `ask_sdk_model.interfaces.amazonpay.response.charge_amazon_pay_result.`**`ChargeAmazonPayR`**

Bases: *`ask_sdk_model.interfaces.amazonpay.v1.charge_amazon_pay_result.ChargeAmazonPayResult`*

Charge Amazon Pay Result Object. It is sent as part of the response to ChargeAmazonPayRequest.

Parameters

- **`amazon_order_reference_id`** (*(optional)* *str*) – The order reference identifier.
- **`authorization_details`** (*(optional)* `ask_sdk_model.interfaces.amazonpay.model.response.authorization_details.AuthorizationDetails`) –

`to_dict`()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.amazonpay.response.setup_amazon_pay_result module

class ask_sdk_model.interfaces.amazonpay.response.setup_amazon_pay_result.**SetupAmazonPayRes**

Bases: object

Setup Amazon Pay Result Object. It is sent as part of the response to SetupAmazonPayRequest.

Parameters billing_agreement_details (*(optional)* ask_sdk_model.interfaces.amazonpay.model.response.billing_agreement_details.BillingAgreementDetails) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.audioplayer.audio_item module

class ask_sdk_model.interfaces.audioplayer.audio_item.**AudioItem**(*stream=None*, *metadata=None*)

Bases: object

Parameters

- **stream** (*(optional)* ask_sdk_model.interfaces.audioplayer.stream.Stream) –
- **metadata** (*(optional)* ask_sdk_model.interfaces.audioplayer.audio_item_metadata.AudioItemMetadata) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.audio_item_metadata module

class ask_sdk_model.interfaces.audioplayer.audio_item_metadata.**AudioItemMetadata**(*title=None, subtitle=None, art=None, background_image*

Bases: object

Encapsulates the metadata about an AudioItem.

Parameters

- **title** (*(optional) str*) – An optional title of the audio item.
- **subtitle** (*(optional) str*) – An optional subtitle of the audio item.
- **art** (*(optional)* ask_sdk_model.interfaces.display.image.Image) – An optional cover art image for the audio item.
- **background_image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) – An optional background image for the audio item.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.audio_player_interface module

class ask_sdk_model.interfaces.audioplayer.audio_player_interface.**AudioPlayerInterface**

Bases: object

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.audio_player_state module

class ask_sdk_model.interfaces.audioplayer.audio_player_state.**AudioPlayerState**(*offset_in_millisec token=None, player_activity=*

Bases: object

Parameters

- **offset_in_milliseconds** (*(optional) int*) –
- **token** (*(optional) str*) –
- **player_activity** (*(optional)* ask_sdk_model.interfaces.audioplayer.player_activity.PlayerActivity) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.clear_behavior module

class ask_sdk_model.interfaces.audioplayer.clear_behavior.**ClearBehavior**
Bases: enum.Enum

Allowed enum values: [CLEAR_ALL, CLEAR_ENQUEUED]

ask_sdk_model.interfaces.audioplayer.clear_queue_directive module

class ask_sdk_model.interfaces.audioplayer.clear_queue_directive.**ClearQueueDirective**(*clear_be*
Bases: *ask_sdk_model.directive.Directive*

Parameters clear_behavior (*(optional)* ask_sdk_model.interfaces.audioplayer.clear_behavior.ClearBehavior) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.current_playback_state module

class ask_sdk_model.interfaces.audioplayer.current_playback_state.**CurrentPlaybackState**(*offset*
playe
token=
Bases: object

Parameters

- **offset_in_milliseconds** (*(optional) int*) –
- **player_activity** (*(optional)* ask_sdk_model.interfaces.audioplayer.player_activity.PlayerActivity) –
- **token** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.error module

class ask_sdk_model.interfaces.audioplayer.error.**Error**(*message=None*, *object_type=None*)
Bases: object

Parameters

- **message** (*(optional) str*) –
- **object_type** (*(optional)* ask_sdk_model.interfaces.audioplayer.error_type.ErrorType) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.error_type module

class ask_sdk_model.interfaces.audioplayer.error_type.**ErrorType**
Bases: enum.Enum

Allowed enum values: [MEDIA_ERROR_INTERNAL_DEVICE_ERROR, MEDIA_ERROR_INTERNAL_SERVER_ERROR, MEDIA_ERROR_INVALID_REQUEST, MEDIA_ERROR_SERVICE_UNAVAILABLE, MEDIA_ERROR_UNKNOWN]

ask_sdk_model.interfaces.audioplayer.play_behavior module

class ask_sdk_model.interfaces.audioplayer.play_behavior.**PlayBehavior**
Bases: enum.Enum

Allowed enum values: [ENQUEUE, REPLACE_ALL, REPLACE_ENQUEUED]

ask_sdk_model.interfaces.audioplayer.play_directive module

class ask_sdk_model.interfaces.audioplayer.play_directive.**PlayDirective**(*play_behavior=None, audio_item=None*)
Bases: *ask_sdk_model.directive.Directive*

Parameters

- **play_behavior** (*(optional)* ask_sdk_model.interfaces.audioplayer.play_behavior.PlayBehavior) –
- **audio_item** (*(optional)* ask_sdk_model.interfaces.audioplayer.audio_item.AudioItem) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_failed_request module

class ask_sdk_model.interfaces.audioplayer.playback_failed_request.**PlaybackFailedRequest**(*re*
tir
ta
lo
ca
cu
re
er
ro
to
ke

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **current_playback_state** (*(optional)* ask_sdk_model.interfaces.audioplayer.current_playback_state.CurrentPlaybackState) –
- **error** (*(optional)* ask_sdk_model.interfaces.audioplayer.error.Error) –
- **token** (*(optional) str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_finished_request module

class ask_sdk_model.interfaces.audioplayer.playback_finished_request.**PlaybackFinishedReques**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.

- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **offset_in_milliseconds** (*(optional) int*) –
- **token** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_nearly_finished_request module

class ask_sdk_model.interfaces.audioplayer.playback_nearly_finished_request.**PlaybackNearlyF**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **offset_in_milliseconds** (*(optional) int*) –
- **token** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_started_request module

class ask_sdk_model.interfaces.audioplayer.playback_started_request.**PlaybackStartedRequest**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **offset_in_milliseconds** (*(optional) int*) –
- **token** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.playback_stopped_request module

class ask_sdk_model.interfaces.audioplayer.playback_stopped_request.**PlaybackStoppedRequest**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

- **offset_in_milliseconds** (*(optional) int*) –
- **token** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.player_activity module

class ask_sdk_model.interfaces.audioplayer.player_activity.**PlayerActivity**
Bases: enum.Enum

Allowed enum values: [PLAYING, PAUSED, FINISHED, BUFFER_UNDERRUN, IDLE, STOPPED]

ask_sdk_model.interfaces.audioplayer.stop_directive module

class ask_sdk_model.interfaces.audioplayer.stop_directive.**StopDirective**
Bases: *ask_sdk_model.directive.Directive*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.audioplayer.stream module

class ask_sdk_model.interfaces.audioplayer.stream.**Stream**(*expected_previous_token=None, token=None, url=None, offset_in_milliseconds=None*)
Bases: object

Parameters

- **expected_previous_token** (*(optional) str*) –
- **token** (*(optional) str*) –
- **url** (*(optional) str*) –
- **offset_in_milliseconds** (*(optional) int*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections package

Subpackages

ask_sdk_model.interfaces.connections.entities package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

---

ask_sdk_model.interfaces.connections.entities.base_entity module

class `ask_sdk_model.interfaces.connections.entities.base_entity.`**`BaseEntity`**(*object_type=None*, *version=None*)

Bases: `object`

Parameters

- **object_type** (*(optional)* `str`) –
- **version** (*(optional)* `str`) – version of the request

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `@type` variable.

Restaurant: *ask_sdk_model.interfaces.connections.entities.restaurant.Restaurant*,

PostalAddress: *ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress*

---

classmethod get_real_child_model(*data*)

Returns the real base class specified by the discriminator

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.connections.entities.postal_address module

class ask_sdk_model.interfaces.connections.entities.postal_address.**PostalAddress**(*version=None, street_address lo- cal- ity=None, re- gion=None, postal_code= coun- try=None*)

Bases: *ask_sdk_model.interfaces.connections.entities.base_entity.BaseEntity*

Postal Address

Parameters

- **version** (*(optional) str*) – version of the request
- **street_address** (*(optional) str*) – street address
- **locality** (*(optional) str*) – locality/city
- **region** (*(optional) str*) – state/region
- **postal_code** (*(optional) str*) – postal/zip code
- **country** (*(optional) str*) – country

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.entities.restaurant module

class ask_sdk_model.interfaces.connections.entities.restaurant.**Restaurant**(*version=None, name=None, location=None*)

Bases: *ask_sdk_model.interfaces.connections.entities.base_entity.BaseEntity*

Restaurant entity

Parameters

- **version** (*(optional) str*) – version of the request
- **name** (*(optional) str*) – name of the restaurant
- **location** (*(optional) ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress*) – location

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.requests package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.connections.requests.base_request module

class ask_sdk_model.interfaces.connections.requests.base_request.**BaseRequest**(*object_type=None*, *version=None*)

Bases: `object`

Parameters

- **object_type** (*(optional) str*) –
- **version** (*(optional) str*) – version of the request

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `@type` variable.

ScheduleFoodEstablishmentReservationRequest: *ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservation_request.ScheduleFoodEstablishmentReservationRequest*,

PrintPDFRequest: *ask_sdk_model.interfaces.connections.requests.print_pdf_request.PrintPDFRequest*,

PrintImageRequest: *ask_sdk_model.interfaces.connections.requests.print_image_request.PrintImageRequest*,

ScheduleTaxiReservationRequest: *ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request.ScheduleTaxiReservationRequest*,

PrintWebPageRequest: *ask_sdk_model.interfaces.connections.requests.print_web_page_request.PrintWebPageRequest*

classmethod get_real_child_model(*data*)

Returns the real base class specified by the discriminator

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.connections.requests.print_image_request module

class ask_sdk_model.interfaces.connections.requests.print_image_request.**PrintImageRequest**(*v*
t
t
u
d
s
t
i
d

Bases: *ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest*

Payload Request object for PrintImage

Parameters

- **version** (*(optional) str*) – version of the request
- **title** (*(optional) str*) – title of the image
- **url** (*(optional) str*) – url of the image
- **description** (*(optional) str*) – description of the image
- **image_type** (*(optional) str*) – type of the image

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.connections.requests.print_pdf_request module

class ask_sdk_model.interfaces.connections.requests.print_pdf_request.**PrintPDFRequest**(*versior*
title=No
url=N
description=N

Bases: *ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest*

Payload Request object for PrintPDF

Parameters

- **version** (*(optional) str*) – version of the request
- **title** (*(optional) str*) – title of the image
- **url** (*(optional) str*) – url of the image
- **description** (*(optional) str*) – description of the image

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.requests.print_web_page_request module

class ask_sdk_model.interfaces.connections.requests.print_web_page_request.**PrintWebPageRequ**

Bases: *ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest*

Payload Request object for PrintWebPage

Parameters

- **version** (*(optional) str*) – version of the request
- **title** (*(optional) str*) – title of the image
- **url** (*(optional) str*) – url of the image
- **description** (*(optional) str*) – description of the image

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservation_request module

class ask_sdk_model.interfaces.connections.requests.schedule_food_establishment_reservatio

Bases: *ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest*

ScheduleFoodEstablishmentReservationRequest for booking restaurant reservation

Parameters

- **version** (*(optional) str*) – version of the request
- **start_time** (*(optional) str*) – start time of the reservation
- **party_size** (*(optional) str*) – party size
- **restaurant** (*(optional)* ask_sdk_model.interfaces.connections.entities.restaurant.Restaurant) – restaurant

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request module

class ask_sdk_model.interfaces.connections.requests.schedule_taxi_reservation_request.**Sched**

Bases: *ask_sdk_model.interfaces.connections.requests.base_request.BaseRequest*

ScheduleTaxiReservationRequest for booking taxi reservation

Parameters

- **version** (*(optional) str*) – version of the request
- **pickup_time** (*(optional) str*) – pickup time
- **party_size** (*(optional) str*) – party size
- **pickup_location** (*(optional)* ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress) – pick up location
- **drop_off_location** (*(optional)* ask_sdk_model.interfaces.connections.entities.postal_address.PostalAddress) – drop off location

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.connections.connections_request module

class ask_sdk_model.interfaces.connections.connections_request.**ConnectionsRequest**(*request_id=I* *timestamp=None*, *locale=None*, *name=None* *payload=None*)

Bases: *ask_sdk_model.request.Request*

This is the request object that a skill will receive as a result of Connections.SendRequest directive from sender skill.

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **name** (*(optional) str*) – Name of the action sent by the referrer skill.
- **payload** (*(optional) dict(str, object)*) – This is an object sent between the two skills for processing a ConnectionsRequest or ConnectionsResponse. This will always be a valid payload based on Action schema for the requester action.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.connections_response module

class ask_sdk_model.interfaces.connections.connections_response.**ConnectionsResponse**(*request_i* *timestamp=No* *locale=Nor* *status=None* *name=No* *payload=Nor* *token=None*)

Bases: *ask_sdk_model.request.Request*

This is the request object that a skill will receive as a result of Connections.SendResponse directive from referrer skill.

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **status** (*(optional) ask_sdk_model.interfaces.connections.connections_status.ConnectionsStatus*) –
- **name** (*(optional) str*) – Name of the action for which response is received.
- **payload** (*(optional) dict(str, object)*) – This is an object sent from referrer skill as is.
- **token** (*(optional) str*) – This is the token that the skill originally sent with the ConnectionsSendRequest directive.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.connections.connections_status module

class ask_sdk_model.interfaces.connections.connections_status.**ConnectionsStatus**(*code=None*, *message=None*)

Bases: object

Connection Status indicates a high level understanding of the result of ConnectionsRequest.

Parameters

- **code** (*(optional) str*) – This is a code signifying the status of the request sent by the skill. Protocol adheres to HTTP status codes.
- **message** (*(optional) str*) – This is a message that goes along with response code that can provide more information about what occurred

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.connections.send_request_directive module

class ask_sdk_model.interfaces.connections.send_request_directive.**SendRequestDirective**(*name*
payload=
token=

Bases: *ask_sdk_model.directive.Directive*

This is the directive that a skill can send as part of their response to a session based request to execute a predefined Connections. This will also return a result to the referring skill. (No Guarantee response will be returned)

Parameters

- **name** (*(optional) str*) – This defines the name of the Connection skill is trying to execute. It must be a valid and supported Connection name.
- **payload** (*(optional) dict(str, object)*) – This is an object sent between the two skills for processing a ConnectionsRequest or ConnectionsResponse. The contract for the object is based on the schema of the Action used in the SendRequestDirective. Invalid payloads will result in errors sent back to the referrer.
- **token** (*(optional) str*) – This is an echo back string that skills send when during Connections.SendRequest directive. They will receive it when they get the ConnectionsResponse. It is never sent to the skill handling the request.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.connections.send_response_directive module

class ask_sdk_model.interfaces.connections.send_response_directive.**SendResponseDirective**(*st*
pa
lo

Bases: *ask_sdk_model.directive.Directive*

This is the directive that a skill can send as part of their response to a session based request to return a response to ConnectionsRequest.

Parameters

- **status** (*(optional) ask_sdk_model.interfaces.connections.connections_status.ConnectionsStatus*) –
- **payload** (*(optional) dict(str, object)*) – This is an object sent to referrer skill as is.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.display.back_button_behavior module

class `ask_sdk_model.interfaces.display.back_button_behavior.`**`BackButtonBehavior`**

Bases: `enum.Enum`

Allowed enum values: [HIDDEN, VISIBLE]

ask_sdk_model.interfaces.display.body_template1 module

class `ask_sdk_model.interfaces.display.body_template1.`**`BodyTemplate1`**(*token=None, back_button=None, background_image=None, title=None, text_content=None*)

Bases: *`ask_sdk_model.interfaces.display.template.Template`*

Parameters

- **token** (*(optional)* `str`) –
- **back_button** (*(optional)* `ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior`) –
- **background_image** (*(optional)* `ask_sdk_model.interfaces.display.image.Image`) –
- **title** (*(optional)* `str`) –
- **text_content** (*(optional)* `ask_sdk_model.interfaces.display.text_content.TextContent`) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.display.body_template2 module

class `ask_sdk_model.interfaces.display.body_template2.`**`BodyTemplate2`**(*token=None, back_button=None, background_image=None, image=None, title=None, text_content=None*)

Bases: *`ask_sdk_model.interfaces.display.template.Template`*

Parameters

- **token** (*(optional) str*) –
- **back_button** (*(optional)* ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior) –
- **background_image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –
- **image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –
- **title** (*(optional) str*) –
- **text_content** (*(optional)* ask_sdk_model.interfaces.display.text_content.TextContent) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.body_template3 module

class ask_sdk_model.interfaces.display.body_template3.**BodyTemplate3**(*token=None, back_button=None, background_image=None, image=None, title=None, text_content=None*)

Bases: *ask_sdk_model.interfaces.display.template.Template*

Parameters

- **token** (*(optional) str*) –
- **back_button** (*(optional)* ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior) –
- **background_image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –
- **image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –
- **title** (*(optional) str*) –
- **text_content** (*(optional)* ask_sdk_model.interfaces.display.text_content.TextContent) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.body_template6 module

class ask_sdk_model.interfaces.display.body_template6.**BodyTemplate6**(*token=None*, *back_button=None*, *background_image=None*, *text_content=None*, *image=None*)

Bases: *ask_sdk_model.interfaces.display.template.Template*

Parameters

- **token** (*(optional) str*) –
- **back_button** (*(optional)* ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior) –
- **background_image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –
- **text_content** (*(optional)* ask_sdk_model.interfaces.display.text_content.TextContent) –
- **image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.display.body_template7 module

class ask_sdk_model.interfaces.display.body_template7.**BodyTemplate7**(*token=None*, *back_button=None*, *title=None*, *image=None*, *background_image=None*)

Bases: *ask_sdk_model.interfaces.display.template.Template*

Parameters

- **token** (*(optional) str*) –
- **back_button** (*(optional)* ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior) –
- **title** (*(optional) str*) –
- **image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –
- **background_image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.display.display_interface module

class ask_sdk_model.interfaces.display.display_interface.**DisplayInterface**(*template_version=None, markup_version=None*)

Bases: object

Parameters

- **template_version** (*(optional) str*) –
- **markup_version** (*(optional) str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.display.display_state module

class ask_sdk_model.interfaces.display.display_state.**DisplayState**(*token=None*)

Bases: object

Parameters **token** (*(optional) str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.display.element_selected_request module

class ask_sdk_model.interfaces.display.element_selected_request.**ElementSelectedRequest**(*reque timestamp locale= token=*

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **token** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.hint module

class ask_sdk_model.interfaces.display.hint.**Hint**(*object_type=None*)
Bases: object

Parameters object_type (*(optional) str*) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

PlainText: *ask_sdk_model.interfaces.display.plain_text_hint.PlainTextHint*

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.hint_directive module

class ask_sdk_model.interfaces.display.hint_directive.**HintDirective**(*hint=None*)
Bases: *ask_sdk_model.directive.Directive*

Parameters hint (*(optional)* ask_sdk_model.interfaces.display.hint.Hint) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.image module

class ask_sdk_model.interfaces.display.image.**Image**(*content_description=None, sources=None*)
Bases: object

Parameters

- **content_description** (*(optional) str*) –
- **sources** (*(optional) list[ask_sdk_model.interfaces.display.image_instance.ImageInstance]*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.image_instance module

class ask_sdk_model.interfaces.display.image_instance.**ImageInstance**(*url=None*, *size=None*, *width_pixels=None*, *height_pixels=None*)

Bases: object

Parameters

- **url** (*(optional) str*) –
- **size** (*(optional) ask_sdk_model.interfaces.display.image_size.ImageSize*) –
- **width_pixels** (*(optional) int*) –
- **height_pixels** (*(optional) int*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.image_size module

class ask_sdk_model.interfaces.display.image_size.**ImageSize**
Bases: enum.Enum

Allowed enum values: [X_SMALL, SMALL, MEDIUM, LARGE, X_LARGE]

ask_sdk_model.interfaces.display.list_item module

class ask_sdk_model.interfaces.display.list_item.**ListItem**(*token=None*, *image=None*, *text_content=None*)

Bases: object

Parameters

- **token** (*(optional) str*) –
- **image** (*(optional) ask_sdk_model.interfaces.display.image.Image*) –
- **text_content** (*(optional) ask_sdk_model.interfaces.display.text_content.TextContent*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.list_template1 module

class ask_sdk_model.interfaces.display.list_template1.**ListTemplate1**(*token=None, back_button=None, background_image=None, title=None, list_items=None*)

Bases: *ask_sdk_model.interfaces.display.template.Template*

Parameters

- **token** (*(optional) str*) –
- **back_button** (*(optional)* ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior) –
- **background_image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –
- **title** (*(optional) str*) –
- **list_items** (*(optional) list[*ask_sdk_model.interfaces.display.list_item.ListItem*]*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.list_template2 module

class ask_sdk_model.interfaces.display.list_template2.**ListTemplate2**(*token=None, back_button=None, background_image=None, title=None, list_items=None*)

Bases: *ask_sdk_model.interfaces.display.template.Template*

Parameters

- **token** (*(optional) str*) –
- **back_button** (*(optional)* ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior) –
- **background_image** (*(optional)* ask_sdk_model.interfaces.display.image.Image) –

- **title** (*(optional) str*) –
- **list_items** (*(optional) list[ask_sdk_model.interfaces.display.list_item.ListItem]*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.plain_text module

class ask_sdk_model.interfaces.display.plain_text.**PlainText**(*text=None*)
Bases: *ask_sdk_model.interfaces.display.text_field.TextField*

Parameters text (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.plain_text_hint module

class ask_sdk_model.interfaces.display.plain_text_hint.**PlainTextHint**(*text=None*)
Bases: *ask_sdk_model.interfaces.display.hint.Hint*

Parameters text (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.render_template_directive module

class ask_sdk_model.interfaces.display.render_template_directive.**RenderTemplateDirective**(*te*
Bases: *ask_sdk_model.directive.Directive*

Parameters template (*(optional) ask_sdk_model.interfaces.display.template.Template*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.rich_text module

class ask_sdk_model.interfaces.display.rich_text.**RichText**(*text=None*)
Bases: *ask_sdk_model.interfaces.display.text_field.TextField*

Parameters **text** (*(optional)* *str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.template module

class ask_sdk_model.interfaces.display.template.**Template**(*object_type=None*, *token=None*, *back_button=None*)
Bases: object

Parameters

- **object_type** (*(optional)* *str*) –
- **token** (*(optional)* *str*) –
- **back_button** (*(optional)* ask_sdk_model.interfaces.display.back_button_behavior.BackButtonBehavior) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

ListTemplate2: *ask_sdk_model.interfaces.display.list_template2.ListTemplate2*,

ListTemplate1: *ask_sdk_model.interfaces.display.list_template1.ListTemplate1*,

BodyTemplate7: *ask_sdk_model.interfaces.display.body_template7.BodyTemplate7*,

BodyTemplate6: *ask_sdk_model.interfaces.display.body_template6.BodyTemplate6*,

BodyTemplate3: *ask_sdk_model.interfaces.display.body_template3.BodyTemplate3*,

BodyTemplate2: *ask_sdk_model.interfaces.display.body_template2.BodyTemplate2*,

BodyTemplate1: *ask_sdk_model.interfaces.display.body_template1.BodyTemplate1*

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.display.text_content module

class ask_sdk_model.interfaces.display.text_content.**TextContent**(*primary_text=None*, *secondary_text=None*, *tertiary_text=None*)

Bases: object

Parameters

- **primary_text** ((*optional*) ask_sdk_model.interfaces.display.text_field.TextField) –
- **secondary_text** ((*optional*) ask_sdk_model.interfaces.display.text_field.TextField) –
- **tertiary_text** ((*optional*) ask_sdk_model.interfaces.display.text_field.TextField) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.display.text_field module

class ask_sdk_model.interfaces.display.text_field.**TextField**(*object_type=None*)

Bases: object

Parameters **object_type** ((*optional*) *str*) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

RichText: *ask_sdk_model.interfaces.display.rich_text.RichText*,

PlainText: *ask_sdk_model.interfaces.display.plain_text.PlainText*

classmethod get_real_child_model(*data*)

Returns the real base class specified by the discriminator

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.geolocation.access module

class `ask_sdk_model.interfaces.geolocation.access.`**`Access`**

Bases: `enum.Enum`

A string representing if Alexa has access to location services running on the hostOS of device.

Allowed enum values: [ENABLED, DISABLED, UNKNOWN]

ask_sdk_model.interfaces.geolocation.altitude module

class `ask_sdk_model.interfaces.geolocation.altitude.`**`Altitude`**(*altitude_in_meters=None, accuracy_in_meters=None*)

Bases: `object`

An object containing the altitude information of the device.

Parameters

- **`altitude_in_meters`** (*(optional) float*) – A double representing the altitude of the device in meters.
- **`accuracy_in_meters`** (*(optional) float*) – A double representing the accuracy of the altitude measurement in meters.

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.coordinate module

class `ask_sdk_model.interfaces.geolocation.coordinate.`**`Coordinate`**(*latitude_in_degrees=None, longitude_in_degrees=None, accuracy_in_meters=None*)

Bases: `object`

An object containing the location information of the device.

Parameters

- **latitude_in_degrees** (*(optional) float*) – A double representing the latitude in degrees of the device.
- **longitude_in_degrees** (*(optional) float*) – A double representing the longitude in degrees of the device.
- **accuracy_in_meters** (*(optional) float*) – A double representing the accuracy of geolocation data in meters.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.geolocation_interface module

class ask_sdk_model.interfaces.geolocation.geolocation_interface.**GeolocationInterface**

Bases: object

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.geolocation_state module

class ask_sdk_model.interfaces.geolocation.geolocation_state.**GeolocationState**(*timestamp=None, coordinate=None, altitude=None, heading=None, speed=None, location_services=Nor*

Bases: object

Parameters

- **timestamp** (*(optional) str*) – Specifies the time when the geolocation data was last collected on the device.
- **coordinate** (*(optional)* ask_sdk_model.interfaces.geolocation.coordinate.Coordinate) –
- **altitude** (*(optional)* ask_sdk_model.interfaces.geolocation.altitude.Altitude) –
- **heading** (*(optional)* ask_sdk_model.interfaces.geolocation.heading.Heading) –

- **speed** (*(optional)* ask_sdk_model.interfaces.geolocation.speed.Speed) –
- **location_services** (*(optional)* ask_sdk_model.interfaces.geolocation.location_services.LocationServices) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.heading module

class ask_sdk_model.interfaces.geolocation.heading.**Heading**(*direction_in_degrees=None*, *accuracy_in_degrees=None*)

Bases: object

An object containing the heading direction information of the device.

Parameters

- **direction_in_degrees** (*(optional)* *float*) – A double representing the direction of the device in degrees.
- **accuracy_in_degrees** (*(optional)* *float*) – A double representing the accuracy of the heading measurement in degrees.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.location_services module

class ask_sdk_model.interfaces.geolocation.location_services.**LocationServices**(*status=None*, *access=None*)

Bases: object

An object containing status and access.

Parameters

- **status** (*(optional)* ask_sdk_model.interfaces.geolocation.status.Status) – A string representing the status of whether location services is currently running or not on the host OS of device.
- **access** (*(optional)* ask_sdk_model.interfaces.geolocation.access.Access) – A string representing if Alexa has access to location services running on the hostOS of device.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.speed module

class ask_sdk_model.interfaces.geolocation.speed.**Speed**(*speed_in_meters_per_second=None*, *accuracy_in_meters_per_second=None*)

Bases: `object`

An object containing the speed information of the device.

Parameters

- **speed_in_meters_per_second** (*(optional)* `float`) – A double representing the speed of the device in meters.
- **accuracy_in_meters_per_second** (*(optional)* `float`) – A double representing the accuracy of the speed measurement in meters.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.geolocation.status module

class ask_sdk_model.interfaces.geolocation.status.**Status**

Bases: `enum.Enum`

A string representing the status of whether location services is currently running or not on the host OS of device.

Allowed enum values: [RUNNING, STOPPED]

ask_sdk_model.interfaces.gadget_controller package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.gadget_controller.set_light_directive module

class ask_sdk_model.interfaces.gadget_controller.set_light_directive.**SetLightDirective**(*versi* *target_g* *parameters=*

Bases: `ask_sdk_model.directive.Directive`

Sends Alexa a command to modify the behavior of connected Echo Buttons.

Parameters

- **version** (*(optional) int*) – The version of the directive. Must be set to 1.
- **target_gadgets** (*(optional) list[str]*) – The gadget IDs that will receive the command. An empty array, or leaving this parameter out, signifies that all gadgets will receive the command.
- **parameters** (*(optional)* ask_sdk_model.services.gadget_controller.set_light_parameters.SetLightParameters) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.game_engine package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.game_engine.input_handler_event_request module

class ask_sdk_model.interfaces.game_engine.input_handler_event_request.**InputHandlerEventRec**

Bases: *ask_sdk_model.request.Request*

Sent when the conditions of an Echo Button event that your skill defined were met.

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

- **originating_request_id** (*(optional) str*) – The corresponding identifier of the request that started the input handler.
- **events** (*(optional) list[ask_sdk_model.services.game_engine.input_handler_event.InputHandlerEvent]*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.game_engine.start_input_handler_directive module

class ask_sdk_model.interfaces.game_engine.start_input_handler_directive.**StartInputHandlerI**

Bases: *ask_sdk_model.directive.Directive*

Parameters

- **timeout** (*(optional) int*) – The maximum run time for this Input Handler, in milliseconds. Although this parameter is required, you can specify events with conditions on which to end the Input Handler earlier.
- **proxies** (*(optional) list[str]*) – Names for unknown gadget IDs to use in recognizers, allocated on a first-come, first-served basis.
- **recognizers** (*(optional) dict(str, ask_sdk_model.services.game_engine.recognizer.Recognizer)*) – Conditions that, at any moment, are either true or false. You use recognizers when you specify the conditions under which your skill is notified of Echo Button input.
- **events** (*(optional) dict(str, ask_sdk_model.services.game_engine.event.Event)*) – The logic that determines when your skill is notified of Echo Button input. Events are listed here as object keys, where the keys specify the name of an event.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.game_engine.stop_input_handler_directive module

class ask_sdk_model.interfaces.game_engine.stop_input_handler_directive.**StopInputHandlerDi**
Bases: *ask_sdk_model.directive.Directive*

Parameters **originating_request_id** (*(optional) str*) – The &#x60;requestId&#x60; of the request that started the input handler.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.messaging package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.messaging.message_received_request module

class ask_sdk_model.interfaces.messaging.message_received_request.**MessageReceivedRequest**(*re*
tir
ta
lo
ca
me
sa

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **message** (*(optional) dict(str, object)*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.monetization package

Subpackages

ask_sdk_model.interfaces.monetization.v1 package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

---

ask_sdk_model.interfaces.monetization.v1.in_skill_product module

class `ask_sdk_model.interfaces.monetization.v1.in_skill_product.`**InSkillProduct**(*product_id=Non*

Bases: `object`

Entity to define In Skill Product over which actions will be performed.

Parameters **product_id** (*(optional)* `str`) – The product ID of In Skill Product.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.monetization.v1.purchase_result module

class `ask_sdk_model.interfaces.monetization.v1.purchase_result.`**PurchaseResult**

Bases: `enum.Enum`

Response from purchase directives: * ACCEPTED - User have accepted the offer to purchase the product * DECLINED - User have declined the offer to purchase the product * NOT_ENTITLED - User tries to cancel/return a product he/she is not entitled to. * ALREADY_PURCHASED - User has already purchased the product * ERROR - An internal error occurred

Allowed enum values: [ACCEPTED, DECLINED, NOT_ENTITLED, ERROR, ALREADY_PURCHASED]

ask_sdk_model.interfaces.playbackcontroller package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.playbackcontroller.next_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.next_command_issued_request.**NextCommandIs**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.playbackcontroller.pause_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.pause_command_issued_request.**PauseCommand**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.playbackcontroller.play_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.play_command_issued_request.**PlayCommandIs**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.playbackcontroller.previous_command_issued_request module

class ask_sdk_model.interfaces.playbackcontroller.previous_command_issued_request.**Previous**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.system package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.system.error module

class `ask_sdk_model.interfaces.system.error.`**`Error`**(*object_type=None, message=None*)

Bases: `object`

Parameters

- **object_type** (*(optional)* `ask_sdk_model.interfaces.system.error_type.ErrorType`) –
- **message** (*(optional)* `str`) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.system.error_cause module

class `ask_sdk_model.interfaces.system.error_cause.`**`ErrorCause`**(*request_id=None*)

Bases: `object`

Parameters **request_id** (*(optional)* `str`) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.system.error_type module

class `ask_sdk_model.interfaces.system.error_type.`**`ErrorType`**

Bases: `enum.Enum`

Allowed enum values: [INVALID_RESPONSE, DEVICE_COMMUNICATION_ERROR, INTERNAL_SERVICE_ERROR]

ask_sdk_model.interfaces.system.exception_encountered_request module

class ask_sdk_model.interfaces.system.exception_encountered_request.**ExceptionEncounteredRec**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **error** (*(optional)* ask_sdk_model.interfaces.system.error.Error) –
- **cause** (*(optional)* ask_sdk_model.interfaces.system.error_cause.ErrorCause) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.system.system_state module

class ask_sdk_model.interfaces.system.system_state.**SystemState**(*application=None*, *user=None*, *device=None*, *api_endpoint=None*, *api_access_token=None*)

Bases: object

Parameters

- **application** (*(optional)* ask_sdk_model.application.Application) –
- **user** (*(optional)* ask_sdk_model.user.User) –
- **device** (*(optional)* ask_sdk_model.device.Device) –
- **api_endpoint** (*(optional) str*) –
- **api_access_token** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.videoapp package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.videoapp.launch_directive module

class ask_sdk_model.interfaces.videoapp.launch_directive.**LaunchDirective**(*video_item=None*)

Bases: *ask_sdk_model.directive.Directive*

Parameters **video_item** (*(optional)* ask_sdk_model.interfaces.videoapp.video_item.VideoItem) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.videoapp.metadata module

class ask_sdk_model.interfaces.videoapp.metadata.**Metadata**(*title=None*, *subtitle=None*)

Bases: object

Parameters

- **title** (*(optional)* str) –
- **subtitle** (*(optional)* str) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.interfaces.videoapp.video_app_interface module

class ask_sdk_model.interfaces.videoapp.video_app_interface.**VideoAppInterface**

Bases: object

to_dict()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.videoapp.video_item module

class ask_sdk_model.interfaces.videoapp.video_item.**VideoItem**(*source=None*, *metadata=None*)
Bases: object

Parameters

- **source** (*(optional) str*) –
- **metadata** (*(optional) ask_sdk_model.interfaces.videoapp.metadata.Metadata*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.viewport package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.interfaces.viewport.experience module

class ask_sdk_model.interfaces.viewport.experience.**Experience**(*arc_minute_width=None*, *arc_minute_height=None*, *can_rotate=None*, *can_resize=None*)
Bases: object

An experience represents a viewing mode used to interact with the device.

Parameters

- **arc_minute_width** (*(optional) float*) – The number of horizontal arc minutes the viewport occupies in the user's visual field when viewed within this experience.
- **arc_minute_height** (*(optional) float*) – The number of vertical arc minutes the viewport occupies in the user's visual field when viewed within this experience.
- **can_rotate** (*(optional) bool*) – Indicates if the viewport can be rotated through 90 degrees.
- **can_resize** (*(optional) bool*) – Indicates if the viewport can be resized, limiting the area which can be used to render the APL response.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.interfaces.viewport.keyboard module

class ask_sdk_model.interfaces.viewport.keyboard.**Keyboard**
Bases: enum.Enum

Represents a physical button input mechanism which can be used to interact with elements shown on the viewport.

Allowed enum values: [DIRECTION]

ask_sdk_model.interfaces.viewport.shape module

class ask_sdk_model.interfaces.viewport.shape.**Shape**
Bases: enum.Enum

The shape of the viewport.

Allowed enum values: [RECTANGLE, ROUND]

ask_sdk_model.interfaces.viewport.touch module

class ask_sdk_model.interfaces.viewport.touch.**Touch**
Bases: enum.Enum

Represents a type of touch input suppported by the device.

Allowed enum values: [SINGLE]

ask_sdk_model.interfaces.viewport.viewport_state module

class ask_sdk_model.interfaces.viewport.viewport_state.**ViewportState**(*experiences=None*, *shape=None*, *pixel_width=None*, *pixel_height=None*, *dpi=None*, *current_pixel_width=None*, *current_pixel_height=None*, *touch=None*, *keyboard=None*)
Bases: object

This object contains the characteristics related to the device's viewport.

Parameters

- **experiences** (*(optional) list[ask_sdk_model.interfaces.viewport.experience.Experience]*) – The experiences supported by the device, in descending order of arcMinuteWidth and arcMinuteHeight.
- **shape** (*(optional) ask_sdk_model.interfaces.viewport.shape.Shape*) –
- **pixel_width** (*(optional) float*) – The number of pixels present in the viewport at its maximum width.
- **pixel_height** (*(optional) float*) – The number of pixels present in the viewport at its maximum height.
- **dpi** (*(optional) float*) – The pixel density of the viewport.
- **current_pixel_width** (*(optional) float*) – The number of horizontal pixels in the viewport that are currently available for Alexa to render an experience.
- **current_pixel_height** (*(optional) float*) – The number of vertical pixels in the viewport that are currently available for Alexa to render an experience.
- **touch** (*(optional) list[ask_sdk_model.interfaces.viewport.touch.Touch]*) – The types of touch supported by the device. An empty array indicates no touch support.
- **keyboard** (*(optional) list[ask_sdk_model.interfaces.viewport.keyboard.Keyboard]*) – The physical button input mechanisms supported by the device. An empty array indicates physical button input is unsupported.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

Module contents

ask_sdk_model.services package

Subpackages

ask_sdk_model.services.device_address package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.device_address.address module

class ask_sdk_model.services.device_address.address.**Address**(*address_line1=None, address_line2=None, address_line3=None, country_code=None, state_or_region=None, city=None, district_or_county=None, postal_code=None*)

Bases: `object`

Represents the full address response from the service.

Parameters

- **address_line1** (*(optional) str*) –
- **address_line2** (*(optional) str*) –
- **address_line3** (*(optional) str*) –
- **country_code** (*(optional) str*) –
- **state_or_region** (*(optional) str*) –
- **city** (*(optional) str*) –
- **district_or_county** (*(optional) str*) –
- **postal_code** (*(optional) str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.device_address.device_address_service_client module

class ask_sdk_model.services.device_address.device_address_service_client.**DeviceAddressServ**

Bases: *ask_sdk_model.services.base_service_client.BaseServiceClient*

ServiceClient for calling the DeviceAddressService APIs.

Parameters api_configuration (`ask_sdk_model.services.api_configuration.ApiConfiguration`) – Instance of *ask_sdk_model.services.api_configuration.ApiConfiguration*

get_country_and_postal_code(*device_id, **kwargs*)

Gets the country and postal code of a device

Parameters device_id (`str`) – (required) The device Id for which to get the country and postal code

Return type Union[*ShortAddress*, *Error*]

get_full_address(*device_id, **kwargs*)

Gets the address of a device

Parameters **device_id** (*str*) – (required) The device Id for which to get the address

Return type Union[*Address*, *Error*]

ask_sdk_model.services.device_address.error module

class ask_sdk_model.services.device_address.error.**Error**(*object_type=None*, *message=None*)

Bases: object

Parameters

- **object_type** (*(optional) str*) – The corresponding type of the http status code being returned.
- **message** (*(optional) str*) – A human readable description of error.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.device_address.short_address module

class ask_sdk_model.services.device_address.short_address.**ShortAddress**(*country_code=None*, *postal_code=None*)

Bases: object

Parameters

- **country_code** (*(optional) str*) –
- **postal_code** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.directive package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.directive.directive module

class ask_sdk_model.services.directive.directive.**Directive**(*object_type=None*)
Bases: object

Parameters **object_type** (*(optional) str*) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets type variable.

VoicePlayer.Speak: *ask_sdk_model.services.directive.speak_directive.SpeakDirective*

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.directive.directive_service_client module

class ask_sdk_model.services.directive.directive_service_client.**DirectiveServiceClient**(*api_c*
Bases: *ask_sdk_model.services.base_service_client.BaseServiceClient*

ServiceClient for calling the DirectiveService APIs.

Parameters **api_configuration** (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of *ask_sdk_model.services.api_configuration.ApiConfiguration*

enqueue(*send_directive_request*, ***kwargs*)
Send directives to Alexa.

Parameters **send_directive_request** (ask_sdk_model.services.directive.send_directive_request.SendDirectiveRequest) – (required) Represents the request object to send in the payload.

Return type None

ask_sdk_model.services.directive.error module

class ask_sdk_model.services.directive.error.**Error**(*code=None*, *message=None*)
Bases: object

Parameters

- **code** (*(optional) int*) – error code to find more information in developer.amazon.com.
- **message** (*(optional) str*) – Readable description of error.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.directive.header module

class ask_sdk_model.services.directive.header.**Header**(*request_id=None*)
Bases: object

Parameters **request_id** (*(optional) str*) – This represents the current requestId for what the skill/speechlet was invoked.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.directive.send_directive_request module

class ask_sdk_model.services.directive.send_directive_request.**SendDirectiveRequest**(*header=No directive=None*)
Bases: object

Send Directive Request payload.

Parameters

- **header** (*(optional)* ask_sdk_model.services.directive.header.Header) – contains the header attributes of the send directive request.
- **directive** (*(optional)* ask_sdk_model.services.directive.directive.Directive) – Directive Content.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.directive.speak_directive module

class ask_sdk_model.services.directive.speak_directive.**SpeakDirective**(*speech=None*)
Bases: *ask_sdk_model.services.directive.directive.Directive*

Parameters **speech** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.gadget_controller package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.gadget_controller.animation_step module

class `ask_sdk_model.services.gadget_controller.animation_step.`**`AnimationStep`**(*duration_ms=None*, *color=None*, *blend=None*)

Bases: `object`

Parameters

- **duration_ms** (*(optional) int*) – The duration in milliseconds to render this step.
- **color** (*(optional) str*) – The color to render specified in RGB hexadecimal values. There are a number of Node.js libraries available for working with color.
- **blend** (*(optional) bool*) – A boolean that indicates whether to interpolate from the previous color into this one over the course of this directive's durationMs.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.gadget_controller.light_animation module

class `ask_sdk_model.services.gadget_controller.light_animation.`**`LightAnimation`**(*repeat=None*, *target_lights=None*, *sequence=None*)

Bases: `object`

Parameters

- **repeat** (*(optional) int*) – The number of times to play this animation.
- **target_lights** (*(optional) list[str]*) – An array of strings that represent the light addresses on the target gadgets that this animation will be applied to. Because the Echo Button has one light only, use [&quot;1&quot;] to signify that this animation should be sent to light one.
- **sequence** (*(optional) list[ask_sdk_model.services.gadget_controller.animation_step.AnimationStep]*) – The animation steps to render in order. The maximum number of steps that you can define is 38. The minimum is 0. Each step must have the following fields, all of which are required.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.gadget_controller.set_light_parameters module

class ask_sdk_model.services.gadget_controller.set_light_parameters.**SetLightParameters**(*trigg trigger_e animations*

Bases: object

Arguments that pertain to animating the buttons.

Parameters

- **trigger_event** (*(optional)* ask_sdk_model.services.gadget_controller.trigger_event_type.TriggerEventType) –
- **trigger_event_time_ms** (*(optional)* *int*) –
- **animations** (*(optional)* *list*[ask_sdk_model.services.gadget_controller.light_animation.LightAnimation]) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.gadget_controller.trigger_event_type module

class ask_sdk_model.services.gadget_controller.trigger_event_type.**TriggerEventType**
Bases: enum.Enum

The action that triggers the animation. Possible values are as follows * &#x60;buttonDown&#x60; - Play the animation when the button is pressed. * &#x60;buttonUp&#x60; - Play the animation when the button is released. * &#x60;none&#x60; - Play the animation as soon as it arrives.

Allowed enum values: [buttonDown, buttonUp, none]

ask_sdk_model.services.game_engine package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.game_engine.deviation_recognizer module

class ask_sdk_model.services.game_engine.deviation_recognizer.**DeviationRecognizer**(*recognizer=*

Bases: *ask_sdk_model.services.game_engine.recognizer.Recognizer*

The deviation recognizer returns true when another specified recognizer reports that the player has deviated from its expected pattern.

Parameters **recognizer** (*(optional) str*) – The name of the recognizer that defines a pattern that must not be deviated from.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.game_engine.event module

class ask_sdk_model.services.game_engine.event.**Event**(*should_end_input_handler=None, meets=None, fails=None, reports=None, maximum_invocations=None, trigger_time_milliseconds=None*)

Bases: object

The events object is where you define the conditions that must be met for your skill to be notified of Echo Button input. You must define at least one event.

Parameters

- **should_end_input_handler** (*(optional) bool*) – Whether the Input Handler should end after this event fires. If true, the Input Handler will stop and no further events will be sent to your skill unless you call StartInputHandler again.
- **meets** (*(optional) list[str]*) –
- **fails** (*(optional) list[str]*) –
- **reports** (*(optional) ask_sdk_model.services.game_engine.event_reporting_type.EventReportingType*) –
- **maximum_invocations** (*(optional) int*) – Enables you to limit the number of times that the skill is notified about the same event during the course of the Input Handler. The default value is 1. This property is mutually exclusive with triggerTimeMilliseconds.
- **trigger_time_milliseconds** (*(optional) int*) – Adds a time constraint to the event. Instead of being considered whenever a raw button event occurs, an event that has this parameter will only be considered once at triggerTimeMilliseconds after the Input Handler has started. Because a time-triggered event can only fire once, the maximumInvocations value is ignored. Omit this property entirely if you do not want to time-constrain the event.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.game_engine.event_reporting_type module

class ask_sdk_model.services.game_engine.event_reporting_type.**EventReportingType**

Bases: `enum.Enum`

Specifies what raw button presses to put in the inputEvents field of the event. * history - All button presses since this Input Handler was started. * matches - Just the button presses that contributed to this event (that is, were in the recognizers). To receive no raw button presses, leave this array empty or do not specify it at all.

Allowed enum values: [history, matches]

ask_sdk_model.services.game_engine.input_event module

class ask_sdk_model.services.game_engine.input_event.**InputEvent**(*gadget_id=None, timestamp=None, action=None, color=None, feature=None*)

Bases: `object`

Parameters

- **gadget_id** (*(optional)* `str`) – The identifier of the Echo Button in question. It matches the gadgetId that you will have discovered in roll call.
- **timestamp** (*(optional)* `str`) – The event's original moment of occurrence, in ISO format.
- **action** (*(optional)* `ask_sdk_model.services.game_engine.input_event_action_type.InputEventActionType`) –
- **color** (*(optional)* `str`) – The hexadecimal RGB values of the button LED at the time of the event.
- **feature** (*(optional)* `str`) – For gadgets with multiple features, this is the feature that the event represents. Echo Buttons have one feature only, so this is always &#x60;press&#x60;.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.game_engine.input_event_action_type module

class ask_sdk_model.services.game_engine.input_event_action_type.**InputEventActionType**

Bases: `enum.Enum`

Either &quot;down&quot; for a button pressed or &quot;up&quot; for a button released.

Allowed enum values: [down, up]

ask_sdk_model.services.game_engine.input_handler_event module

class ask_sdk_model.services.game_engine.input_handler_event.**InputHandlerEvent**(*name=None*, *input_events=None*

Bases: object

Parameters

- **name** (*(optional) str*) – The name of the event as you defined it in your GameEngine.StartInputHandler directive.
- **input_events** (*(optional) list[ask_sdk_model.services.game_engine.input_event.InputEvent]*) – A chronologically ordered report of the raw Button Events that contributed to this Input Handler Event.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.game_engine.pattern module

class ask_sdk_model.services.game_engine.pattern.**Pattern**(*gadget_ids=None*, *colors=None*, *action=None*, *repeat=None*)

Bases: object

An object that provides all of the events that need to occur, in a specific order, for this recognizer to be true. Omitting any parameters in this object means &quot;match anything&quot;.

Parameters

- **gadget_ids** (*(optional) list[str]*) – A whitelist of gadgetIds that are eligible for this match.
- **colors** (*(optional) list[str]*) – A whitelist of colors that are eligible for this match.
- **action** (*(optional) ask_sdk_model.services.game_engine.input_event_action_type.InputEventActionType*) –
- **repeat** (*(optional) int*) – The number of times that the specified action must occur to be considered complete.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.game_engine.pattern_recognizer module

class ask_sdk_model.services.game_engine.pattern_recognizer.**PatternRecognizer**(*anchor=None*, *fuzzy=None*, *gadget_ids=None*, *actions=None*, *pattern=None*)

Bases: *ask_sdk_model.services.game_engine.recognizer.Recognizer*

This recognizer is true when all of the specified events have occurred in the specified order.

Parameters

- **anchor** (*(optional)* ask_sdk_model.services.game_engine.pattern_recognizer_anchor_type.PatternRecognizerAnchorType) –
- **fuzzy** (*(optional)* *bool*) – When true, the recognizer will ignore additional events that occur between the events specified in the pattern.
- **gadget_ids** (*(optional)* *list[str]*) – The gadget IDs of the Echo Buttons to consider in this pattern recognizer.
- **actions** (*(optional)* *list[str]*) – The actions to consider in this pattern recognizer. All other actions will be ignored.
- **pattern** (*(optional)* *list*[ask_sdk_model.services.game_engine.pattern.Pattern]) – An object that provides all of the events that need to occur, in a specific order, for this recognizer to be true. Omitting any parameters in this object means &quot;match anything&quot;.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.game_engine.pattern_recognizer_anchor_type module

class ask_sdk_model.services.game_engine.pattern_recognizer_anchor_type.**PatternRecognizerAn**

Bases: enum.Enum

Where the pattern must appear in the history of this input handler. * &#x60;start&#x60; - (Default) The first event in the pattern must be the first event in the history of raw Echo Button events. * &#x60;end&#x60; - The last event in the pattern must be the last event in the history of raw Echo Button events. * &#x60;anywhere&#x60; - The pattern may appear anywhere in the history of raw Echo Button events.

Allowed enum values: [start, end, anywhere]

ask_sdk_model.services.game_engine.progress_recognizer module

class ask_sdk_model.services.game_engine.progress_recognizer.**ProgressRecognizer**(*recognizer=Non*
completion=None)

Bases: *ask_sdk_model.services.game_engine.recognizer.Recognizer*

This recognizer consults another recognizer for the degree of completion, and is true if that degree is above the specified threshold. The completion parameter is specified as a decimal percentage.

Parameters

- **recognizer** (*(optional) str*) – The name of a recognizer for which to track the progress.
- **completion** (*(optional) float*) – The completion threshold, as a decimal percentage, of the specified recognizer before which this recognizer becomes true.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.game_engine.recognizer module

class ask_sdk_model.services.game_engine.recognizer.**Recognizer**(*object_type=None*)
Bases: object

Recognizers are conditions that, at any moment, are either true or false, based on all the raw button events that the Input Handler has received in the time elapsed since the Input Handler session started.

Parameters **object_type** (*(optional) str*) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

progress: *ask_sdk_model.services.game_engine.progress_recognizer.ProgressRecognizer*,

match: *ask_sdk_model.services.game_engine.pattern_recognizer.PatternRecognizer*,

deviation: *ask_sdk_model.services.game_engine.deviation_recognizer.DeviationRecognizer*

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.list_management.alexa_list module

class ask_sdk_model.services.list_management.alexa_list.**AlexaList**(*list_id=None*, *name=None*, *state=None*, *version=None*, *items=None*, *links=None*)

Bases: `object`

Parameters

- **list_id** (*(optional)* `str`) –
- **name** (*(optional)* `str`) –
- **state** (*(optional)* `ask_sdk_model.services.list_management.list_state.ListState`) –
- **version** (*(optional)* `int`) –
- **items** (*(optional)* `list`*[*`ask_sdk_model.services.list_management.alexa_list_item.AlexaListItem`*]*) –
- **links** (*(optional)* `ask_sdk_model.services.list_management.links.Links`) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.alexa_list_item module

class ask_sdk_model.services.list_management.alexa_list_item.**AlexaListItem**(*id=None*, *version=None*, *value=None*, *status=None*, *created_time=None*, *updated_time=None*, *href=None*)

Bases: `object`

Parameters

- **id** (*(optional) str*) –
- **version** (*(optional) int*) –
- **value** (*(optional) str*) –
- **status** (*(optional)* ask_sdk_model.services.list_management.list_item_state.ListItemState) –
- **created_time** (*(optional) str*) –
- **updated_time** (*(optional) str*) –
- **href** (*(optional) str*) – URL to retrieve the item from.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.alexa_list_metadata module

class ask_sdk_model.services.list_management.alexa_list_metadata.**AlexaListMetadata**(*list_id=No*
name=Non
state=None
version=None
status_map=N

Bases: `object`

Parameters

- **list_id** (*(optional) str*) –
- **name** (*(optional) str*) –
- **state** (*(optional)* ask_sdk_model.services.list_management.list_state.ListState) –
- **version** (*(optional) int*) –

- **status_map** (*(optional)* *list*[ask_sdk_model.services.list_management.status.Status]) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.alexa_lists_metadata module

class ask_sdk_model.services.list_management.alexa_lists_metadata.**AlexaListsMetadata**(*lists=No*

Bases: object

Parameters lists (*(optional)* *list*[ask_sdk_model.services.list_management.alexa_list_metadata.AlexaListMetadata]) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.create_list_item_request module

class ask_sdk_model.services.list_management.create_list_item_request.**CreateListItemReques**

Bases: object

Parameters

- **value** (*(optional)* *str*) –
- **status** (*(optional)* ask_sdk_model.services.list_management.list_item_state.ListItemState) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.create_list_request module

class ask_sdk_model.services.list_management.create_list_request.**CreateListRequest**(*name=Non*
state=None

Bases: object

Parameters

- **name** (*(optional)* *str*) –
- **state** (*(optional)* ask_sdk_model.services.list_management.list_state.ListState) –

to_dict()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.error module

class ask_sdk_model.services.list_management.error.**Error**(*object_type=None*, *message=None*)
Bases: object

Parameters

- **object_type** (*(optional) str*) –
- **message** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.forbidden_error module

class ask_sdk_model.services.list_management.forbidden_error.**ForbiddenError**(*message=None*)
Bases: object

Parameters **message** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.links module

class ask_sdk_model.services.list_management.links.**Links**(*next=None*)
Bases: object

Parameters **next** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.list_body module

class ask_sdk_model.services.list_management.list_body.**ListBody**(*list_id=None*)
Bases: object

Parameters **list_id** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.list_created_event_request module

class ask_sdk_model.services.list_management.list_created_event_request.**ListCreatedEventRec**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.services.list_management.list_body.ListBody) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.list_deleted_event_request module

class ask_sdk_model.services.list_management.list_deleted_event_request.**ListDeletedEventRec**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.

- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional) ask_sdk_model.services.list_management.list_body.ListBody*) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.list_item_body module

class ask_sdk_model.services.list_management.list_item_body.**ListItemBody**(*list_id=None*, *list_item_ids=None*)

Bases: object

Parameters

- **list_id** (*(optional) str*) –
- **list_item_ids** (*(optional) list[str]*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.list_item_state module

class ask_sdk_model.services.list_management.list_item_state.**ListItemState**

Bases: enum.Enum

Allowed enum values: [active, completed]

ask_sdk_model.services.list_management.list_items_created_event_request module

class ask_sdk_model.services.list_management.list_items_created_event_request.**ListItemsCrea**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional) ask_sdk_model.services.list_management.list_item_body.ListItemBody*) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.list_items_deleted_event_request module

class ask_sdk_model.services.list_management.list_items_deleted_event_request.**ListItemsDele**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional) ask_sdk_model.services.list_management.list_item_body.ListItemBody*) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.list_management.list_items_updated_event_request module

class ask_sdk_model.services.list_management.list_items_updated_event_request.**ListItemsUpda**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.services.list_management.list_item_body.ListItemBody) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.list_management_service_client module

class ask_sdk_model.services.list_management.list_management_service_client.**ListManagementS**
Bases: *ask_sdk_model.services.base_service_client.BaseServiceClient*

ServiceClient for calling the ListManagementService APIs.

Parameters api_configuration (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of *ask_sdk_model.services.api_configuration.ApiConfiguration*

create_list(*create_list_request*, ***kwargs*)
This API creates a custom list. The new list name must be different than any existing list name.

Parameters create_list_request (ask_sdk_model.services.list_management.create_list_request.CreateListRequest) – (required)

Return type Union[*Error*, *AlexaListMetadata*]

create_list_item(*list_id*, *create_list_item_request*, ***kwargs*)
This API creates an item in an active list or in a default list.

Parameters

- **list_id** (*str*) – (required) The customer's listId retrieved from a getListsMetadata call.
- **create_list_item_request** (ask_sdk_model.services.list_management.create_list_item_request.CreateListItemRequest) – (required)

Return type Union[*AlexaListItem*, *Error*]

delete_list(*list_id*, ***kwargs*)

This API deletes a customer custom list.

Parameters list_id (*str*) – (required) Value of the customer's listId retrieved from a getListsMetadata call

Return type None

delete_list_item(*list_id*, *item_id*, ***kwargs*)

This API deletes an item in the specified list.

Parameters

- **list_id** (*str*) – (required) The customer's listId is retrieved from a getListsMetadata call.
- **item_id** (*str*) – (required) The customer's itemId is retrieved from a GetList call.

Return type None

get_list(*list_id*, *status*, ***kwargs*)

Retrieves the list metadata including the items in the list with requested status.

Parameters

- **list_id** (*str*) – (required) Retrieved from a call to GetListsMetadata to specify the listId in the request path.
- **status** (*str*) – (required) Specify the status of the list.

Return type Union[*AlexaList*, *Error*]

get_list_item(*list_id*, *item_id*, ***kwargs*)

This API can be used to retrieve single item with in any list by listId and itemId. This API can read list items from an archived list. Attempting to read list items from a deleted list return an ObjectNotFound 404 error.

Parameters

- **list_id** (*str*) – (required) Retrieved from a call to getListsMetadata
- **item_id** (*str*) – (required) itemId within a list is retrieved from a getList call

Return type Union[*AlexaListItem*, *Error*]

get_lists_metadata(***kwargs*)

Retrieves the metadata for all customer lists, including the customer's default lists.

Return type Union[*ForbiddenError*, *Error*, *AlexaListsMetadata*]

update_list(*list_id*, *update_list_request*, ***kwargs*)

This API updates a custom list. Only the list name or state can be updated. An Alexa customer can turn an archived list into an active one.

Parameters

- **list_id** (*str*) – (required) Value of the customer's listId retrieved from a getListsMetadata call.

- **update_list_request** (ask_sdk_model.services.list_management.update_list_request.UpdateListRequest) – (required)

Return type Union[*Error*, *AlexaListMetadata*]

update_list_item(*list_id*, *item_id*, *update_list_item_request*, ***kwargs*)
API used to update an item value or item status.

Parameters

- **list_id** (*str*) – (required) Customer's listId
- **item_id** (*str*) – (required) itemId to be updated in the list
- **update_list_item_request** (ask_sdk_model.services.list_management.update_list_item_request.UpdateListItemRequest) – (required)

Return type Union[*AlexaListItem*, *Error*]

ask_sdk_model.services.list_management.list_state module

class ask_sdk_model.services.list_management.list_state.**ListState**
Bases: enum.Enum

Allowed enum values: [active, archived]

ask_sdk_model.services.list_management.list_updated_event_request module

class ask_sdk_model.services.list_management.list_updated_event_request.**ListUpdatedEventReq**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.services.list_management.list_body.ListBody) –
- **event_creation_time** (*(optional) datetime*) –
- **event_publishing_time** (*(optional) datetime*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.status module

class ask_sdk_model.services.list_management.status.**Status**(*url=None*, *status=None*)

Bases: object

Parameters

- **url** (*(optional)* str) –
- **status** (*(optional)* ask_sdk_model.services.list_management.list_item_state.ListItemState) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.update_list_item_request module

class ask_sdk_model.services.list_management.update_list_item_request.**UpdateListItemRequest**

Bases: object

Parameters

- **value** (*(optional)* str) – New item value
- **status** (*(optional)* ask_sdk_model.services.list_management.list_item_state.ListItemState) – Item Status
- **version** (*(optional)* int) – Item version when it was read.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.list_management.update_list_request module

class ask_sdk_model.services.list_management.update_list_request.**UpdateListRequest**(*name=Non*
state=None
version=None

Bases: object

Parameters

- **name** (*(optional)* str) –

- **state** (*(optional)* ask_sdk_model.services.list_management.list_state.ListState) –
- **version** (*(optional)* int) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.monetization package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.monetization.entitled_state module

class ask_sdk_model.services.monetization.entitled_state.**EntitledState**

Bases: enum.Enum

State determining if the user is entitled to the product. Note - Any new values introduced later should be treated as 'NOT_ENTITLED'. * 'ENTITLED' - The user is entitled to the product. * 'NOT_ENTITLED' - The user is not entitled to the product.

Allowed enum values: [ENTITLED, NOT_ENTITLED]

ask_sdk_model.services.monetization.error module

class ask_sdk_model.services.monetization.error.**Error**(*message=None*)

Bases: object

Describes error detail

Parameters **message** (*(optional)* str) – Readable description of error

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.monetization.in_skill_product module

class ask_sdk_model.services.monetization.in_skill_product.**InSkillProduct**(*product_id=None, reference_name=None, name=None, object_type=None, summary=None, purchasable=None, entitled=None, active_entitlement_count=, purchase_mode=None*)

Bases: `object`

Parameters

- **product_id** (*(optional) str*) – Product Id
- **reference_name** (*(optional) str*) – Developer selected in-skill product name. This is for developer reference only.
- **name** (*(optional) str*) – Name of the product in the language from the &quot;Accept-Language&quot; header
- **object_type** (*(optional)* `ask_sdk_model.services.monetization.product_type.ProductType`) –
- **summary** (*(optional) str*) – Product summary in the language from the &quot;Accept-Language&quot; header
- **purchasable** (*(optional)* `ask_sdk_model.services.monetization.purchasable_state.PurchasableState`) –
- **entitled** (*(optional)* `ask_sdk_model.services.monetization.entitled_state.EntitledState`) –
- **active_entitlement_count** (*(optional) int*) – Total active purchases of the product made by the user. Note - For ENTITLEMENT and SUBSCRIPTION product types, the value is either zero(NOT_ENTITLED) or one(ENTITLED). For CONSUMABLE product type the value is zero or more, as CONSUMABLE can be re-purchased.
- **purchase_mode** (*(optional)* `ask_sdk_model.services.monetization.purchase_mode.PurchaseMode`) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.monetization.in_skill_products_response module

class ask_sdk_model.services.monetization.in_skill_products_response.**InSkillProductsRespons**

Bases: object

Parameters

- **in_skill_products** (*(optional) list[ask_sdk_model.services.monetization.in_skill_product.InSkillProduct]*) – List of In-Skill Products
- **is_truncated** (*(optional) bool*) –
- **next_token** (*(optional) str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.monetization.monetization_service_client module

class ask_sdk_model.services.monetization.monetization_service_client.**MonetizationServiceCl**

Bases: *ask_sdk_model.services.base_service_client.BaseServiceClient*

ServiceClient for calling the MonetizationService APIs.

Parameters **api_configuration** (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of *ask_sdk_model.services.api_configuration.ApiConfiguration*

get_in_skill_product(*accept_language*, *product_id*, ***kwargs*)

Get In-Skill Product information based on user context for the Skill.

Parameters

- **accept_language** (*str*) – (required) User's locale/language in context
- **product_id** (*str*) – (required) Product Id.

Return type Union[*Error*, *InSkillProduct*]

get_in_skill_products(*accept_language*, ***kwargs*)

Gets In-Skill Products based on user's context for the Skill.

Parameters

- **accept_language** (*str*) – (required) User's locale/language in context
- **purchasable** (*str*) – Filter products based on whether they are purchasable by the user or not. * 'PURCHASABLE' - Products that are purchasable by the user. * 'NOT_PURCHASABLE' - Products that are not purchasable by the user.
- **entitled** (*str*) – Filter products based on whether they are entitled to the user or not. * 'ENTITLED' - Products that the user is entitled to. * 'NOT_ENTITLED' - Products that the user is not entitled to.

- **product_type** (*str*) – Product type. * ‘SUBSCRIPTION’ - Once purchased, customers will own the content for the subscription period. * ‘ENTITLEMENT’ - Once purchased, customers will own the content forever. * ‘CONSUMABLE’ - Once purchased, customers will be entitled to the content until it is consumed. It can also be re-purchased.
- **next_token** (*str*) – When response to this API call is truncated (that is, isTruncated response element value is true), the response also includes the nextToken element, the value of which can be used in the next request as the continuation-token to list the next set of objects. The continuation token is an opaque value that In-Skill Products API understands. Token has expiry of 24 hours.
- **max_results** (*float*) – sets the maximum number of results returned in the response body. If you want to retrieve fewer than upper limit of 100 results, you can add this parameter to your request. maxResults should not exceed the upper limit. The response might contain fewer results than maxResults, but it will never contain more. If there are additional results that satisfy the search criteria, but these results were not returned because maxResults was exceeded, the response contains isTruncated = true.

Return type Union[*Error*, *InSkillProductsResponse*]

ask_sdk_model.services.monetization.product_type module

class ask_sdk_model.services.monetization.product_type.**ProductType**

Bases: enum.Enum

Product type. * ’SUBSCRIPTION’ - Once purchased, customers will own the content for the subscription period. * ’ENTITLEMENT’ - Once purchased, customers will own the content forever. * ’CONSUMABLE’ - Once purchased, customers will be entitled to the content until it is consumed. It can also be re-purchased.

Allowed enum values: [SUBSCRIPTION, ENTITLEMENT, CONSUMABLE]

ask_sdk_model.services.monetization.purchasable_state module

class ask_sdk_model.services.monetization.purchasable_state.**PurchasableState**

Bases: enum.Enum

State determining if the product is purchasable by the user. Note - Any new values introduced later should be treated as ’NOT_PURCHASABLE’. * ’PURCHASABLE’ - The product is purchasable by the user. * ’NOT_PURCHASABLE’ - The product is not purchasable by the user.

Allowed enum values: [PURCHASABLE, NOT_PURCHASABLE]

ask_sdk_model.services.monetization.purchase_mode module

class ask_sdk_model.services.monetization.purchase_mode.**PurchaseMode**

Bases: enum.Enum

Indicates if the entitlements are for TEST or LIVE purchases. * ’TEST’ - test purchases made by developers or beta testers. Purchase not sent to payment processing. * ’LIVE’ - purchases made by live customers. Purchase sent to payment processing.

Allowed enum values: [TEST, LIVE]

ask_sdk_model.services.reminder_management package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.reminder_management.alert_info module

class `ask_sdk_model.services.reminder_management.alert_info.`**`AlertInfo`**(*spoken_info=None*)

Bases: `object`

Alert info for VUI / GUI

Parameters **spoken_info** (*(optional) ask_sdk_model.services.reminder_management.spoken_info.SpokenInfo*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.alert_info_spoken_info module

class `ask_sdk_model.services.reminder_management.alert_info_spoken_info.`**`SpokenInfo`**(*content=N*

Bases: `object`

Parameters for VUI presentation of the reminder

Parameters **content** (*(optional) list[ask_sdk_model.services.reminder_management.spoken_text.SpokenText]*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.error module

class `ask_sdk_model.services.reminder_management.error.`**`Error`**(*code=None, message=None*)

Bases: `object`

Parameters

- **code** (*(optional) str*) – Domain specific error code
- **message** (*(optional) str*) – Detailed error message

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.reminder_management.event module

class ask_sdk_model.services.reminder_management.event.**Event**(*status=None*, *alert_token=None*)

Bases: `object`

Parameters

- **status** (*(optional)* `ask_sdk_model.services.reminder_management.status.Status`) –
- **alert_token** (*(optional)* `str`) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.reminder_management.get_reminder_response module

class ask_sdk_model.services.reminder_management.get_reminder_response.**GetReminderResponse**

Bases: `ask_sdk_model.services.reminder_management.reminder.Reminder`

Response object for get reminder request

Parameters

- **alert_token** (*(optional)* `str`) – Unique id of this reminder alert
- **created_time** (*(optional)* `datetime`) – Valid ISO 8601 format - Creation time of this reminder alert
- **updated_time** (*(optional)* `datetime`) – Valid ISO 8601 format - Last updated time of this reminder alert
- **status** (*(optional)* `ask_sdk_model.services.reminder_management.status.Status`) –
- **trigger** (*(optional)* `ask_sdk_model.services.reminder_management.trigger.Trigger`) –

- **alert_info** (*(optional)* ask_sdk_model.services.reminder_management.alert_info.AlertInfo) –
- **push_notification** (*(optional)* ask_sdk_model.services.reminder_management.push_notification.PushNotification) –
- **version** (*(optional)* *str*) – Version of reminder alert

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.get_reminders_response module

class ask_sdk_model.services.reminder_management.get_reminders_response.**GetRemindersRespons**

Bases: object

Response object for get reminders request

Parameters

- **total_count** (*(optional)* *str*) – Total count of reminders returned
- **alerts** (*(optional)* *list[ask_sdk_model.services.reminder_management.reminder.Reminder]*) – List of reminders
- **links** (*(optional)* *str*) – Link to retrieve next set of alerts if total count is greater than max results

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.push_notification module

class ask_sdk_model.services.reminder_management.push_notification.**PushNotification**(*status=N*

Bases: object

Enable / disable reminders push notifications to Alexa mobile apps

Parameters **status** (*(optional)* ask_sdk_model.services.reminder_management.push_notification_status.PushNotificationStatus) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.push_notification_status module

class ask_sdk_model.services.reminder_management.push_notification_status.**PushNotificationS**

Bases: enum.Enum

Push notification status - Enabled/Disabled

Allowed enum values: [ENABLED, DISABLED]

ask_sdk_model.services.reminder_management.recurrence module

class ask_sdk_model.services.reminder_management.recurrence.**Recurrence**(*freq=None*, *by_day=None*, *interval=None*)

Bases: object

Recurring date/time using the RFC 5545 standard in JSON object form

Parameters

- **freq** (*(optional)* ask_sdk_model.services.reminder_management.recurrence_freq.RecurrenceFreq) –
- **by_day** (*(optional)* *list*[ask_sdk_model.services.reminder_management.recurrence_day.RecurrenceDay]) –
- **interval** (*(optional)* *int*) – contains a positive integer representing at which intervals the recurrence rule repeats

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.recurrence_day module

class ask_sdk_model.services.reminder_management.recurrence_day.**RecurrenceDay**

Bases: enum.Enum

Day of recurrence

Allowed enum values: [SU, MO, TU, WE, TH, FR, SA]

ask_sdk_model.services.reminder_management.recurrence_freq module

class ask_sdk_model.services.reminder_management.recurrence_freq.**RecurrenceFreq**

Bases: enum.Enum

Frequency of recurrence

Allowed enum values: [WEEKLY, DAILY]

ask_sdk_model.services.reminder_management.reminder module

class ask_sdk_model.services.reminder_management.reminder.**Reminder**(*alert_token=None, created_time=None, updated_time=None, status=None, trigger=None, alert_info=None, push_notification=None, version=None*)

Bases: `object`

Reminder object

Parameters

- **alert_token** (*(optional) str*) – Unique id of this reminder alert
- **created_time** (*(optional) datetime*) – Valid ISO 8601 format - Creation time of this reminder alert
- **updated_time** (*(optional) datetime*) – Valid ISO 8601 format - Last updated time of this reminder alert
- **status** (*(optional)* `ask_sdk_model.services.reminder_management.status.Status`) –
- **trigger** (*(optional)* `ask_sdk_model.services.reminder_management.trigger.Trigger`) –
- **alert_info** (*(optional)* `ask_sdk_model.services.reminder_management.alert_info.AlertInfo`) –
- **push_notification** (*(optional)* `ask_sdk_model.services.reminder_management.push_notification.PushNotification`) –
- **version** (*(optional) str*) – Version of reminder alert

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_created_event_request module

class ask_sdk_model.services.reminder_management.reminder_created_event_request.**ReminderCre**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional) ask_sdk_model.services.reminder_management.event.Event*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_deleted_event module

class ask_sdk_model.services.reminder_management.reminder_deleted_event.**ReminderDeletedEve**

Bases: object

Parameters **alert_tokens** (*(optional) list[str]*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_deleted_event_request module

class ask_sdk_model.services.reminder_management.reminder_deleted_event_request.**ReminderDel**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional) ask_sdk_model.services.reminder_management.reminder_deleted_event.ReminderDeletedEvent*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_management_service_client module

class ask_sdk_model.services.reminder_management.reminder_management_service_client.**Reminde**

Bases: *ask_sdk_model.services.base_service_client.BaseServiceClient*

ServiceClient for calling the ReminderManagementService APIs.

Parameters **api_configuration** (ask_sdk_model.services.api_configuration.ApiConfiguration) – Instance of *ask_sdk_model.services.api_configuration.ApiConfiguration*

create_reminder(*reminder_request*, ***kwargs*)

This API is invoked by the skill to create a new reminder.

Parameters **reminder_request** (ask_sdk_model.services.reminder_management.reminder_request.ReminderRequest) – (required)

Return type Union[*ReminderResponse*, *Error*]

delete_reminder(*alert_token*, ***kwargs*)

This API is invoked by the skill to delete a single reminder.

Parameters **alert_token** (*str*) – (required)

Return type None

delete_reminders(***kwargs*)

This API is invoked by the skill to delete all reminders created by the caller.

Return type None

get_reminder(*alert_token*, ***kwargs*)

This API is invoked by the skill to get a single reminder.

Parameters **alert_token** (*str*) – (required)

Return type Union[*GetReminderResponse*, *Error*]

get_reminders(***kwargs*)

This API is invoked by the skill to get a all reminders created by the caller.

Return type Union[*GetRemindersResponse*, *Error*]

update_reminder(*alert_token*, *reminder_request*, ***kwargs*)

This API is invoked by the skill to update a reminder.

Parameters

- **alert_token** (*str*) – (required)
- **reminder_request** (ask_sdk_model.services.reminder_management.reminder_request.ReminderRequest) – (required)

Return type Union[*ReminderResponse*, *Error*]

ask_sdk_model.services.reminder_management.reminder_request module

class ask_sdk_model.services.reminder_management.reminder_request.**ReminderRequest**(*request_time*
trigger=None,
alert_info=N
push_notific

Bases: `object`

Input request for creating a reminder

Parameters

- **request_time** (*(optional) datetime*) – Valid ISO 8601 format - Creation time of this reminder alert
- **trigger** (*(optional)* `ask_sdk_model.services.reminder_management.trigger.Trigger`) –
- **alert_info** (*(optional)* `ask_sdk_model.services.reminder_management.alert_info.AlertInfo`) –
- **push_notification** (*(optional)* `ask_sdk_model.services.reminder_management.push_notification.PushNotification`) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_response module

class ask_sdk_model.services.reminder_management.reminder_response.**ReminderResponse**(*alert_toke*
created_time
updated_tim
status=None
version=Nor
href=Non

Bases: `object`

Response object for post/put/delete reminder request

Parameters

- **alert_token** (*(optional) str*) – Unique id of this reminder alert
- **created_time** (*(optional) str*) – Valid ISO 8601 format - Creation time of this reminder alert
- **updated_time** (*(optional) str*) – Valid ISO 8601 format - Last updated time of this reminder alert
- **status** (*(optional)* `ask_sdk_model.services.reminder_management.status.Status`) –

- **version** (*(optional) str*) – Version of reminder alert
- **href** (*(optional) str*) – URI to retrieve the created alert

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_started_event_request module

class ask_sdk_model.services.reminder_management.reminder_started_event_request.**ReminderSta**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.services.reminder_management.event.Event) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_status_changed_event_request module

class ask_sdk_model.services.reminder_management.reminder_status_changed_event_request.**Rem**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.

- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.services.reminder_management.event.Event) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.reminder_management.reminder_updated_event_request module

class ask_sdk_model.services.reminder_management.reminder_updated_event_request.**ReminderUpd**

Bases: *ask_sdk_model.request.Request*

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **body** (*(optional)* ask_sdk_model.services.reminder_management.event.Event) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.services.reminder_management.spoken_text module

class ask_sdk_model.services.reminder_management.spoken_text.**SpokenText**(*locale=None*, *ssml=None*, *text=None*)

Bases: object

Parameters

- **locale** (*(optional) str*) – The locale in which the spoken text is rendered. e.g. en-US
- **ssml** (*(optional) str*) – Spoken text in SSML format.
- **text** (*(optional) str*) – Spoken text in plain text format.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.status module

class ask_sdk_model.services.reminder_management.status.**Status**

Bases: enum.Enum

Status of reminder

Allowed enum values: [ON, COMPLETED]

ask_sdk_model.services.reminder_management.trigger module

class ask_sdk_model.services.reminder_management.trigger.**Trigger**(*object_type=None*, *scheduled_time=None*, *offset_in_seconds=None*, *time_zone_id=None*, *recurrence=None*)

Bases: object

Trigger information for Reminder

Parameters

- **object_type** (*(optional)* ask_sdk_model.services.reminder_management.trigger_type.TriggerType) –
- **scheduled_time** (*(optional)* *datetime*) – Valid ISO 8601 format - Intended trigger time
- **offset_in_seconds** (*(optional)* *int*) – If reminder is set using relative time, use this field to specify the time after which reminder ll ring (in seconds)
- **time_zone_id** (*(optional)* *str*) – Intended reminder's timezone
- **recurrence** (*(optional)* ask_sdk_model.services.reminder_management.recurrence.Recurrence) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.reminder_management.trigger_type module

class ask_sdk_model.services.reminder_management.trigger_type.**TriggerType**

Bases: enum.Enum

Type of reminder - Absolute / Relative

Allowed enum values: [SCHEDULED_ABSOLUTE, SCHEDULED_RELATIVE]

ask_sdk_model.services.ups package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.ups.distance_units module

class `ask_sdk_model.services.ups.distance_units.`**`DistanceUnits`**

Bases: `enum.Enum`

Allowed enum values: [METRIC, IMPERIAL]

ask_sdk_model.services.ups.error module

class `ask_sdk_model.services.ups.error.`**`Error`**(*code=None*, *message=None*)

Bases: `object`

Parameters

- **code** (*(optional)* `ask_sdk_model.services.ups.error_code.ErrorCode`) –
- **message** (*(optional)* `str`) – A human readable description of error.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.ups.error_code module

class `ask_sdk_model.services.ups.error_code.`**`ErrorCode`**

Bases: `enum.Enum`

A more precise error code. Some of these codes may not apply to some APIs. - INVALID_KEY: the setting key is not supported - INVALID_VALUE: the setting value is not valid - INVALID_TOKEN: the token is invalid - INVALID_URI: the uri is invalid - DEVICE_UNREACHABLE: the device is offline - UNKNOWN_ERROR: internal service error

Allowed enum values: [INVALID_KEY, INVALID_VALUE, INVALID_TOKEN, INVALID_URI, DEVICE_UNREACHABLE, UNKNOWN_ERROR]

ask_sdk_model.services.ups.phone_number module

class ask_sdk_model.services.ups.phone_number.**PhoneNumber**(*country_code=None*, *phone_number=None*)

Bases: `object`

Parameters

- **country_code** (*(optional)* `str`) –
- **phone_number** (*(optional)* `str`) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.services.ups.ups_service_client module

class ask_sdk_model.services.ups.ups_service_client.**UpsServiceClient**(*api_configuration*)

Bases: *ask_sdk_model.services.base_service_client.BaseServiceClient*

ServiceClient for calling the UpsService APIs.

Parameters **api_configuration** (`ask_sdk_model.services.api_configuration.ApiConfiguration`) – Instance of *ask_sdk_model.services.api_configuration.ApiConfiguration*

get_profile_email(***kwargs*)

Gets the email address of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:email:read]

Return type Union[str, *Error*]

get_profile_given_name(***kwargs*)

Gets the given name (first name) of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:given_name:read]

Return type Union[str, *Error*]

get_profile_mobile_number(***kwargs*)

Gets the mobile phone number of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:mobile_number:read]

Return type Union[*PhoneNumber*, *Error*]

get_profile_name(***kwargs*)

Gets the full name of the customer associated with the current enablement. Requires customer consent for scopes: [alexa::profile:name:read]

Return type Union[str, *Error*]

get_system_distance_units(*device_id*, ***kwargs*)

Gets the distance measurement unit of the device. Does not require explict customer consent.

Parameters **device_id** (`str`) – (required) The device Id

Return type Union[*Error*, *DistanceUnits*]

get_system_temperature_unit(*device_id*, ***kwargs*)

Gets the temperature measurement units of the device. Does not require explict customer consent.

Parameters **device_id** (*str*) – (required) The device Id

Return type Union[*TemperatureUnit*, *Error*]

get_system_time_zone(*device_id*, ***kwargs*)

Gets the time zone of the device. Does not require explict customer consent.

Parameters **device_id** (*str*) – (required) The device Id

Return type Union[str, *Error*]

ask_sdk_model.services.ups.temperature_unit module

class ask_sdk_model.services.ups.temperature_unit.**TemperatureUnit**

Bases: `enum.Enum`

Allowed enum values: [CELSIUS, FAHRENHEIT]

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.services.api_client module

class ask_sdk_model.services.api_client.**ApiClient**

Bases: `object`

Represents a basic contract for API request invocation.

invoke(*request*)

Dispatches a request to an API endpoint described in the request.

The ApiClient is expected to resolve in the case an API returns a non-200 HTTP status code. The responsibility of translating a particular response code to an error lies with the caller.

Parameters **request** (`ApiClientRequest`) – Request to dispatch to the ApiClient

Returns Response from the client call

Return type *ApiClientResponse*

ask_sdk_model.services.api_client_message module

class ask_sdk_model.services.api_client_message.**ApiClientMessage**(*headers=None*, *body=None*)

Bases: `object`

Represents the interface between *`ask_sdk_model.services.api_client.ApiClient`* implementation and a Service Client.

Parameters

- **headers** (*list[tuple[str, str]]*) – List of header tuples
- **body** (*str*) – Body of the message

ask_sdk_model.services.api_client_request module

class ask_sdk_model.services.api_client_request.**ApiClientRequest**(*headers=None, body=None, url=None, method=None*)

Bases: *ask_sdk_model.services.api_client_message.ApiClientMessage*

Represents a request sent from Service Clients to an *ask_sdk_model.services.api_client.ApiClient* implementation.

Parameters

- **headers** (*list[tuple[str, str]]*) – List of header tuples
- **body** (*str*) – Body of the message
- **url** (*str*) – Url of the request
- **method** (*str*) – Method called with the request

ask_sdk_model.services.api_client_response module

class ask_sdk_model.services.api_client_response.**ApiClientResponse**(*headers=None, body=None, status_code=None*)

Bases: *ask_sdk_model.services.api_client_message.ApiClientMessage*

Represents a request sent from Service Clients to an *ask_sdk_model.services.api_client.ApiClient* class. implementation.

Parameters

- **headers** (*list[tuple[str, str]]*) – List of header tuples
- **body** (*str*) – Body of the message
- **status_code** (*int*) – Status code of the response

ask_sdk_model.services.api_configuration module

class ask_sdk_model.services.api_configuration.**ApiConfiguration**(*serializer=None, api_client=None, authorization_value=None, api_endpoint=None*)

Bases: object

Represents a class that provides API configuration options needed by service clients.

Parameters

- **serializer** (*(optional)* `ask_sdk_model.services.serializer.Serializer`) – serializer implementation for encoding/decoding JSON from/to Object models.
- **api_client** (*(optional)* `ask_sdk_model.services.api_client.ApiClient`) – API Client implementation
- **authorization_value** (*(optional)* `str`) – Authorization value to be used on any calls of the service client instance
- **api_endpoint** (*(optional)* `str`) – Endpoint to hit by the service client instance

ask_sdk_model.services.base_service_client module

class ask_sdk_model.services.base_service_client.**BaseServiceClient**(*api_configuration*)

Bases: `object`

Class to be used as the base class for the generated service clients.

The class has to be implemented by the service clients and this class instantiation is not supported

Parameters **api_configuration** (`ask_sdk_model.services.api_configuration.ApiConfiguration`) – ApiConfiguration implementation

invoke(*method*, *endpoint*, *path*, *query_params*, *header_params*, *path_params*, *response_definitions*, *body*, *response_type*)

Calls the ApiClient based on the ServiceClient specific data provided as well as handles the well-known responses from the Api.

Parameters

- **method** (`str`) – Http method
- **endpoint** – Base endpoint to make the request to
- **path** (`str`) – Specific path to hit. It might contain variables to be interpolated with path_params
- **query_params** (`list(tuple(str, str))`) – Parameter values to be sent as part of query string
- **header_params** (`list(tuple(str, str))`) – Parameter values to be sent as headers
- **path_params** (`dict(str, str)`) – Parameter values to be interpolated in the path
- **response_definitions** (`list(ask_sdk_model.services.service_client_response.ServiceClientResponse)`) – Well-known expected responses by the ServiceClient
- **body** (`object`) – Request body
- **response_type** (`class`) – Type of the expected response if applicable

Returns Response object instance of the response_type provided

Return type object

Raises *`ask_sdk_model.services.service_exception.ServiceException`*

ask_sdk_model.services.serializer module

class ask_sdk_model.services.serializer.**Serializer**

Bases: `object`

Represents an abstract object used for Serialization tasks

deserialize(*payload*, *obj_type*)

Deserializes the payload to object of provided obj_type.

Parameters

- **payload** (*str*) – String to deserialize
- **obj_type** (*object*) – Target type of deserialization

Returns Deserialized object

Return type object

serialize(*obj*)

Serializes an object into a string.

Parameters **obj** – object to serialize

Returns serialized object in string format

Return type str

ask_sdk_model.services.service_client_factory module

class ask_sdk_model.services.service_client_factory.**ServiceClientFactory**(*api_configuration*)

Bases: `object`

ServiceClientFactory class to help build service clients.

Parameters **api_configuration** (`ApiConfiguration`) – API Configuration for calling services

get_device_address_service()

Get DeviceAddressServiceClient for device_address_service.

Returns Client for calling the service

Return type *DeviceAddressServiceClient*

Raises `ValueError`

get_directive_service()

Get DirectiveServiceClient for directive_service.

Returns Client for calling the service

Return type *DirectiveServiceClient*

Raises `ValueError`

get_list_management_service()

Get ListManagementServiceClient for list_management_service.

Returns Client for calling the service

Return type *ListManagementServiceClient*

Raises `ValueError`

get_monetization_service()
Get MonetizationServiceClient for monetization_service.

Returns Client for calling the service

Return type *MonetizationServiceClient*

Raises `ValueError`

get_reminder_management_service()
Get ReminderManagementServiceClient for reminder_management_service.

Returns Client for calling the service

Return type *ReminderManagementServiceClient*

Raises `ValueError`

get_ups_service()
Get UpsServiceClient for ups_service.

Returns Client for calling the service

Return type *UpsServiceClient*

Raises `ValueError`

ask_sdk_model.services.service_client_response module

class ask_sdk_model.services.service_client_response.**ServiceClientResponse**(*response_type*, *status_code*, *message*)

Bases: `object`

Represents a well-known response object by Service Client.

Parameters

- **response_type** (*Response class*) – Well-known representation of the response
- **status_code** (*int*) – Status code to be attached to the response
- **message** (*str*) – Message to be attached to the response

ask_sdk_model.services.service_exception module

exception ask_sdk_model.services.service_exception.**ServiceException**(*message*, *status_code*, *headers*, *body*)

Bases: `Exception`

Exception thrown by a Service client when an error response was received or some operation failed.

Parameters

- **message** (*str*) – Description of the error
- **status_code** (*int*) – Status code of the HTTP Response

- **headers** (*list(tuple(str, str))*) – Headers of the Http response that return the failure
- **body** (*object*) – Body of the HTTP Response

ask_sdk_model.slu package

Subpackages

ask_sdk_model.slu.entityresolution package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.slu.entityresolution.resolution module

class `ask_sdk_model.slu.entityresolution.resolution.`**`Resolution`**(*authority=None, status=None, values=None*)

Bases: `object`

Represents a possible authority for entity resolution

Parameters

- **authority** (*(optional) str*) –
- **status** (*(optional)* `ask_sdk_model.slu.entityresolution.status.Status`) –
- **values** (*(optional) list[*`ask_sdk_model.slu.entityresolution.value_wrapper.ValueWrapper`*]*) –

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.slu.entityresolution.resolutions module

class `ask_sdk_model.slu.entityresolution.resolutions.`**`Resolutions`**(*resolutions_per_authority=None*)

Bases: `object`

Represents the results of resolving the words captured from the user's utterance. This is included for slots that use a custom slot type or a built-in slot type that you have extended with your own values. Note that resolutions is not included for built-in slot types that you have not extended.

Parameters **resolutions_per_authority** (*(optional)* *list*[ask_sdk_model.slu.entityresolution.resolution.Resolution]) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.slu.entityresolution.status module

class ask_sdk_model.slu.entityresolution.status.**Status**(*code=None*)
Bases: object

Parameters **code** (*(optional)* ask_sdk_model.slu.entityresolution.status_code.StatusCode) – Indication of the results of attempting to resolve the user utterance against the defined slot types.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.slu.entityresolution.status_code module

class ask_sdk_model.slu.entityresolution.status_code.**StatusCode**
Bases: enum.Enum

Indication of the results of attempting to resolve the user utterance against the defined slot types.

Allowed enum values: [ER_SUCCESS_MATCH, ER_SUCCESS_NO_MATCH, ER_ERROR_TIMEOUT, ER_ERROR_EXCEPTION]

ask_sdk_model.slu.entityresolution.value module

class ask_sdk_model.slu.entityresolution.value.**Value**(*name=None*, *id=None*)
Bases: object

Represents the resolved value for the slot, based on the user's utterance and slot type definition.

Parameters

- **name** (*(optional)* *str*) – The name for the resolution value.
- **id** (*(optional)* *str*) – The id for the resolution value.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.slu.entityresolution.value_wrapper module

class ask_sdk_model.slu.entityresolution.value_wrapper.**ValueWrapper**(*value=None*)

Bases: object

A wrapper class for an entity resolution value used for JSON serialization.

Parameters **value** (*(optional)* ask_sdk_model.slu.entityresolution.value.Value) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.ui package

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.ui.ask_for_permissions_consent_card module

class ask_sdk_model.ui.ask_for_permissions_consent_card.**AskForPermissionsConsentCard**(*permiss*

Bases: *ask_sdk_model.ui.card.Card*

Parameters **permissions** (*(optional)* *list[str]*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.ui.card module

class ask_sdk_model.ui.card.**Card**(*object_type=None*)

Bases: object

Parameters **object_type** (*(optional)* *str*) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

LinkAccount: *ask_sdk_model.ui.link_account_card.LinkAccountCard*, Standard: *ask_sdk_model.ui.standard_card.StandardCard*,

AskForPermissionsConsent: *ask_sdk_model.ui.ask_for_permissions_consent_card.AskForPermissionsConsentCard*,

Simple: *ask_sdk_model.ui.simple_card.SimpleCard*

---

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.image module

class ask_sdk_model.ui.image.**Image**(*small_image_url=None*, *large_image_url=None*)
Bases: object

Parameters

- **small_image_url** (*(optional) str*) –
- **large_image_url** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.link_account_card module

class ask_sdk_model.ui.link_account_card.**LinkAccountCard**
Bases: *ask_sdk_model.ui.card.Card*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.output_speech module

class ask_sdk_model.ui.output_speech.**OutputSpeech**(*object_type=None*, *play_behavior=None*)
Bases: object

Parameters

- **object_type** (*(optional) str*) –

- **play_behavior** (*(optional) ask_sdk_model.ui.play_behavior.PlayBehavior*) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

SSML: *ask_sdk_model.ui.ssml_output_speech.SsmlOutputSpeech*,

PlainText: *ask_sdk_model.ui.plain_text_output_speech.PlainTextOutputSpeech*

---

classmethod get_real_child_model(*data*)
Returns the real base class specified by the discriminator

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.plain_text_output_speech module

class ask_sdk_model.ui.plain_text_output_speech.**PlainTextOutputSpeech**(*play_behavior=None, text=None*)
Bases: *ask_sdk_model.ui.output_speech.OutputSpeech*

Parameters

- **play_behavior** (*(optional) ask_sdk_model.ui.play_behavior.PlayBehavior*) –
- **text** (*(optional) str*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.reprompt module

class ask_sdk_model.ui.reprompt.**Reprompt**(*output_speech=None*)
Bases: `object`

Parameters output_speech (*(optional)* `ask_sdk_model.ui.output_speech.OutputSpeech`) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.ui.simple_card module

class ask_sdk_model.ui.simple_card.**SimpleCard**(*title=None*, *content=None*)

Bases: *ask_sdk_model.ui.card.Card*

Parameters

- **title** (*(optional) str*) –
- **content** (*(optional) str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.ui.ssml_output_speech module

class ask_sdk_model.ui.ssml_output_speech.**SsmlOutputSpeech**(*play_behavior=None*, *ssml=None*)

Bases: *ask_sdk_model.ui.output_speech.OutputSpeech*

Parameters

- **play_behavior** (*(optional) ask_sdk_model.ui.play_behavior.PlayBehavior*) –
- **ssml** (*(optional) str*) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.ui.standard_card module

class ask_sdk_model.ui.standard_card.**StandardCard**(*title=None*, *text=None*, *image=None*)

Bases: *ask_sdk_model.ui.card.Card*

Parameters

- **title** (*(optional) str*) –
- **text** (*(optional) str*) –
- **image** (*(optional)* ask_sdk_model.ui.image.Image) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

Submodules

Note: Canonical imports have been added in the `__init__.py` of the package. This helps in importing the class directly from the package, than through the module.

For eg: if `package a` has `module b` with `class C`, you can do `from a import C` instead of `from a.b import C`.

ask_sdk_model.application module

class `ask_sdk_model.application.`**`Application`**(*application_id=None*)

Bases: `object`

An object containing an application ID. This is used to verify that the request was intended for your service.

Parameters **application_id** (*(optional)* `str`) – A string representing the application identifier for your skill.

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.context module

class `ask_sdk_model.context.`**`Context`**(*system=None*, *audio_player=None*, *display=None*, *geolocation=None*, *viewport=None*)

Bases: `object`

Parameters

- **system** (*(optional)* `ask_sdk_model.interfaces.system.system_state.SystemState`) – Provides information about the current state of the Alexa service and the device interacting with your skill.
- **audio_player** (*(optional)* `ask_sdk_model.interfaces.audioplayer.audio_player_state.AudioPlayerState`) – Provides the current state for the AudioPlayer interface.
- **display** (*(optional)* `ask_sdk_model.interfaces.display.display_state.DisplayState`) – Provides the current state for the Display interface.
- **geolocation** (*(optional)* `ask_sdk_model.interfaces.geolocation.geolocation_state.GeolocationState`) – Provides the last gathered geolocation information of the device.
- **viewport** (*(optional)* `ask_sdk_model.interfaces.viewport.viewport_state.ViewportState`) – Provides the characteristics of a device's viewport.

`to_dict`()

Returns the model properties as a dict

`to_str`()

Returns the string representation of the model

ask_sdk_model.device module

class ask_sdk_model.device.**Device**(*device_id=None*, *supported_interfaces=None*)

Bases: object

An object providing information about the device used to send the request. The device object contains both deviceId and supportedInterfaces properties. The deviceId property uniquely identifies the device. The supportedInterfaces property lists each interface that the device supports. For example, if supportedInterfaces includes AudioPlayer {}, then you know that the device supports streaming audio using the AudioPlayer interface.

Parameters

- **device_id** (*(optional) str*) – The deviceId property uniquely identifies the device.
- **supported_interfaces** (*(optional)* ask_sdk_model.supported_interfaces.SupportedInterfaces) – Lists each interface that the device supports. For example, if supportedInterfaces includes AudioPlayer {}, then you know that the device supports streaming audio using the AudioPlayer interface

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.dialog_state module

class ask_sdk_model.dialog_state.**DialogState**

Bases: enum.Enum

Enumeration indicating the status of the multi-turn dialog. This property is included if the skill meets the requirements to use the Dialog directives. Note that COMPLETED is only possible when you use the Dialog.Delegate directive. If you use intent confirmation, dialogState is considered COMPLETED if the user denies the entire intent (for instance, by answering "no" when asked the confirmation prompt). Be sure to also check the confirmationStatus property on the Intent object before fulfilling the user's request.

Allowed enum values: [STARTED, IN_PROGRESS, COMPLETED]

ask_sdk_model.directive module

class ask_sdk_model.directive.**Directive**(*object_type=None*)

Bases: object

Parameters **object_type** (*(optional) str*) –

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

AudioPlayer.Stop:
ask_sdk_model.interfaces.audioplayer.stop_directive.StopDirective,

Dialog.ConfirmSlot:
ask_sdk_model.dialog.confirm_slot_directive.ConfirmSlotDirective,

AudioPlayer.Play: *ask_sdk_model.interfaces.audioplayer.play_directive.PlayDirective*,

Alexa.Presentation.APL.ExecuteCommands: *ask_sdk_model.interfaces.alexa.presentation.apl.execute_commands_directive.ExecuteCommandsDirective*,

Connections.SendRequest: *ask_sdk_model.interfaces.connections.send_request_directive.SendRequestDirective*,

Display.RenderTemplate: *ask_sdk_model.interfaces.display.render_template_directive.RenderTemplateDirective*,

GadgetController.SetLight: *ask_sdk_model.interfaces.gadget_controller.set_light_directive.SetLightDirective*,

Dialog.Delegate: *ask_sdk_model.dialog.delegate_directive.DelegateDirective*,

Hint: *ask_sdk_model.interfaces.display.hint_directive.HintDirective*,

Dialog.ConfirmIntent: *ask_sdk_model.dialog.confirm_intent_directive.ConfirmIntentDirective*,

GameEngine.StartInputHandler: *ask_sdk_model.interfaces.game_engine.start_input_handler_directive.StartInputHandlerDirective*,

VideoApp.Launch: *ask_sdk_model.interfaces.videoapp.launch_directive.LaunchDirective*,

GameEngine.StopInputHandler: *ask_sdk_model.interfaces.game_engine.stop_input_handler_directive.StopInputHandlerDirective*,

Alexa.Presentation.APL.RenderDocument: *ask_sdk_model.interfaces.alexa.presentation.apl.render_document_directive.RenderDocumentDirective*,

Connections.SendResponse: *ask_sdk_model.interfaces.connections.send_response_directive.SendResponseDirective*,

Dialog.ElicitSlot: *ask_sdk_model.dialog.elicit_slot_directive.ElicitSlotDirective*,

AudioPlayer.ClearQueue: *ask_sdk_model.interfaces.audioplayer.clear_queue_directive.ClearQueueDirective*

---

classmethod get_real_child_model(*data*)

Returns the real base class specified by the discriminator

to_dict()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.intent module

class ask_sdk_model.intent.**Intent**(*name=None*, *slots=None*, *confirmation_status=None*)
Bases: `object`

An object that represents what the user wants.

Parameters

- **name** (*(optional) str*) – A string representing the name of the intent.
- **slots** (*(optional) dict(str, ask_sdk_model.slot.Slot)*) – A map of key-value pairs that further describes what the user meant based on a predefined intent schema. The map can be empty.
- **confirmation_status** (*(optional) ask_sdk_model.intent_confirmation_status.IntentConfirmationStatus*) –

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.intent_confirmation_status module

class ask_sdk_model.intent_confirmation_status.**IntentConfirmationStatus**
Bases: `enum.Enum`

Indication of whether an intent or slot has been explicitly confirmed or denied by the user, or neither.

Allowed enum values: [NONE, DENIED, CONFIRMED]

ask_sdk_model.intent_request module

class ask_sdk_model.intent_request.**IntentRequest**(*request_id=None*, *timestamp=None*, *locale=None*, *dialog_state=None*, *intent=None*)
Bases: *ask_sdk_model.request.Request*

An IntentRequest is an object that represents a request made to a skill based on what the user wants to do.

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user’s locale. For example: en-US. This value is only provided with certain request types.

- **dialog_state** (*(optional)* ask_sdk_model.dialog_state.DialogState) – Enumeration indicating the status of the multi-turn dialog. This property is included if the skill meets the requirements to use the Dialog directives. Note that COMPLETED is only possible when you use the Dialog.Delegate directive. If you use intent confirmation, dialogState is considered COMPLETED if the user denies the entire intent (for instance, by answering "no" when asked the confirmation prompt). Be sure to also check the confirmationStatus property on the Intent object before fulfilling the user's request.
- **intent** (*(optional)* ask_sdk_model.intent.Intent) – An object that represents what the user wants.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.launch_request module

class ask_sdk_model.launch_request.**LaunchRequest**(*request_id=None*, *timestamp=None*, *locale=None*)

Bases: *ask_sdk_model.request.Request*

Represents that a user made a request to an Alexa skill, but did not provide a specific intent.

Parameters

- **request_id** (*(optional)* *str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional)* *datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional)* *str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.permissions module

class ask_sdk_model.permissions.**Permissions**(*consent_token=None*, *scopes=None*)

Bases: object

Contains a consentToken allowing the skill access to information that the customer has consented to provide, such as address information. Note that the consentToken is deprecated. Use the apiAccessToken available in the context object to determine the user's permissions.

Parameters

- **consent_token** (*(optional)* *str*) – A token listing all the permissions granted for this user.

- **scopes** (*(optional) dict(str, ask_sdk_model.scope.Scope)*) – A map where the key is a LoginWithAmazon(LWA) scope and value is a list of key:value pairs which describe the state of user actions on the LWA scope. For e.g. &quot;scopes&quot; :{ &quot;alexa::devices:all:geolocation:read&quot;:{&quot;status&quot;:&quot;GRANTED&quot;}} This value of &quot;alexa::devices:all:geolocation:read&quot; will determine if the Geolocation data access is granted by the user, or else it will show a card of type AskForPermissionsConsent to the user to get this permission.

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

ask_sdk_model.request module

class ask_sdk_model.request.**Request**(*object_type=None*, *request_id=None*, *timestamp=None*, *locale=None*)

Bases: object

A request object that provides the details of the user's request. The request body contains the parameters necessary for the service to perform its logic and generate a response.

Parameters

- **object_type** (*(optional) str*) – Describes the type of the request.
- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.

Note: This is an abstract class. Use the following mapping, to figure out the model class to be instantiated, that sets `type` variable.

AudioPlayer.PlaybackFinished: *ask_sdk_model.interfaces.audioplayer.playback_finished_request.PlaybackFinishedRequest*,

AlexaSkillEvent.SkillEnabled: *ask_sdk_model.events.skillevents.skill_enabled_request.SkillEnabledRequest*,

AlexaHouseholdListEvent.ListUpdated: *ask_sdk_model.services.list_management.list_updated_event_request.ListUpdatedEventRequest*,

Alexa.Presentation.APL.UserEvent: *ask_sdk_model.interfaces.alexa.presentation.apl.user_event.UserEvent*,

AlexaSkillEvent.SkillDisabled: *ask_sdk_model.events.skillevents.skill_disabled_request.SkillDisabledRequest*, Display.ElementSelected: *ask_sdk_model.interfaces.display.element_selected_request.ElementSelectedRequest*,

AlexaSkillEvent.SkillPermissionChanged: *ask_sdk_model.events.skillevents.permission_changed_request.PermissionChangedRequest*,

AlexaHouseholdListEvent.ItemsCreated: *ask_sdk_model.services.list_management.list_items_created_event_request.ListItemsCreatedEventRequest*,

Reminders.ReminderUpdated: *ask_sdk_model.services.reminder_management.reminder_updated_event_request.ReminderUpdatedEventRequest*,

SessionEndedRequest: *ask_sdk_model.session_ended_request.SessionEndedRequest*,

IntentRequest: *ask_sdk_model.intent_request.IntentRequest*,

AudioPlayer.PlaybackFailed: *ask_sdk_model.interfaces.audioplayer.playback_failed_request.PlaybackFailedRequest*,

CanFulfillIntentRequest: *ask_sdk_model.canfulfill.can_fulfill_intent_request.CanFulfillIntentRequest*,

Reminders.ReminderStarted: *ask_sdk_model.services.reminder_management.reminder_started_event_request.ReminderStartedEventRequest*,

LaunchRequest: *ask_sdk_model.launch_request.LaunchRequest*,

Reminders.ReminderCreated: *ask_sdk_model.services.reminder_management.reminder_created_event_request.ReminderCreatedEventRequest*,

AudioPlayer.PlaybackStopped: *ask_sdk_model.interfaces.audioplayer.playback_stopped_request.PlaybackStoppedRequest*,

PlaybackController.PreviousCommandIssued: *ask_sdk_model.interfaces.playbackcontroller.previous_command_issued_request.PreviousCommandIssuedRequest*,

AlexaHouseholdListEvent.ItemsUpdated: *ask_sdk_model.services.list_management.list_items_updated_event_request.ListItemsUpdatedEventRequest*,

AlexaSkillEvent.SkillAccountLinked: *ask_sdk_model.events.skillevents.account_linked_request.AccountLinkedRequest*,

AlexaHouseholdListEvent.ListCreated: *ask_sdk_model.services.list_management.list_created_event_request.ListCreatedEventRequest*,

AudioPlayer.PlaybackStarted: *ask_sdk_model.interfaces.audioplayer.playback_started_request.PlaybackStartedRequest*, AudioPlayer.PlaybackNearlyFinished: *ask_sdk_model.interfaces.audioplayer.playback_nearly_finished_request.PlaybackNearlyFinishedRequest*,

Reminders.ReminderStatusChanged: *ask_sdk_model.services.reminder_management.reminder_status_changed_event_request.ReminderStatusChangedEventRequest*,

AlexaHouseholdListEvent.ItemsDeleted: *ask_sdk_model.services.list_management.list_items_deleted_event_request.ListItemsDeletedEventRequest*,

Reminders.ReminderDeleted: *ask_sdk_model.services.reminder_management.reminder_deleted_event_request.ReminderDeletedEventRequest*,

Connections.Response: *ask_sdk_model.interfaces.connections.connections_response.ConnectionsResponse*,

Messaging.MessageReceived: *ask_sdk_model.interfaces.messaging.message_received_request.MessageReceivedRequest*,

Connections.Request: *ask_sdk_model.interfaces.connections.connections_request.ConnectionsRequest*,

System.ExceptionEncountered: *ask_sdk_model.interfaces.system.exception_encountered_request.ExceptionEncounteredRequest*,

AlexaSkillEvent.SkillPermissionAccepted: *ask_sdk_model.events.skillevents.permission_accepted_request.PermissionAcceptedRequest*,

AlexaHouseholdListEvent.ListDeleted: *ask_sdk_model.services.list_management.list_deleted_event_request.ListDeletedEventRequest*,

GameEngine.InputHandlerEvent: *ask_sdk_model.interfaces.game_engine.input_handler_event_request.InputHandlerEventRequest*,

PlaybackController.NextCommandIssued: *ask_sdk_model.interfaces.playbackcontroller.next_command_issued_request.NextCommandIssuedRequest*,

PlaybackController.PauseCommandIssued: *ask_sdk_model.interfaces.playbackcontroller.pause_command_issued_request.PauseCommandIssuedRequest*,

PlaybackController.PlayCommandIssued: *ask_sdk_model.interfaces.playbackcontroller.play_command_issued_request.PlayCommandIssuedRequest*

---

classmethod get_real_child_model(*data*)

Returns the real base class specified by the discriminator

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.request_envelope module

class ask_sdk_model.request_envelope.**RequestEnvelope**(*version=None*, *session=None*, *context=None*, *request=None*)

Bases: object

Request wrapper for all requests sent to your Skill.

Parameters

- **version** (*(optional) str*) – The version specifier for the request.
- **session** (*(optional)* ask_sdk_model.session.Session) – The session object provides additional context associated with the request.
- **context** (*(optional)* ask_sdk_model.context.Context) – The context object provides your skill with information about the current state of the Alexa service and device at the time the request is sent to your service. This is included on all requests. For requests sent in the context of a session (LaunchRequest and IntentRequest), the context object duplicates the user and application information that is also available in the session.
- **request** (*(optional)* ask_sdk_model.request.Request) – A request object that provides the details of the user's request.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.response module

class ask_sdk_model.response.**Response**(*output_speech=None*, *card=None*, *reprompt=None*, *directives=None*, *should_end_session=None*, *can_fulfill_intent=None*)

Bases: object

Parameters

- **output_speech** (*(optional)* ask_sdk_model.ui.output_speech.OutputSpeech) –
- **card** (*(optional)* ask_sdk_model.ui.card.Card) –
- **reprompt** (*(optional)* ask_sdk_model.ui.reprompt.Reprompt) –
- **directives** (*(optional) list[*ask_sdk_model.directive.Directive*]*) –
- **should_end_session** (*(optional) bool*) –
- **can_fulfill_intent** (*(optional)* ask_sdk_model.canfulfill.can_fulfill_intent.CanFulfillIntent) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.response_envelope module

class ask_sdk_model.response_envelope.**ResponseEnvelope**(*version=None*, *session_attributes=None*, *user_agent=None*, *response=None*)

Bases: object

Parameters

- **version** (*(optional)* *str*) –
- **session_attributes** (*(optional)* *dict(str, object)*) –
- **user_agent** (*(optional)* *str*) –
- **response** (*(optional)* ask_sdk_model.response.Response) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.session module

class ask_sdk_model.session.**Session**(*new=None*, *session_id=None*, *user=None*, *attributes=None*, *application=None*)

Bases: object

Represents a single execution of the alexa service

Parameters

- **new** (*(optional)* *bool*) – A boolean value indicating whether this is a new session. Returns true for a new session or false for an existing session.
- **session_id** (*(optional)* *str*) – A string that represents a unique identifier per a user's active session.
- **user** (*(optional)* ask_sdk_model.user.User) – An object that describes the user making the request.
- **attributes** (*(optional)* *dict(str, object)*) – A map of key-value pairs. The attributes map is empty for requests where a new session has started with the property new set to true. When returning your response, you can include data you need to persist during the session in the sessionAttributes property. The attributes you provide are then passed back to your skill on the next request.
- **application** (*(optional)* ask_sdk_model.application.Application) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.session_ended_error module

class ask_sdk_model.session_ended_error.**SessionEndedError**(*object_type=None*, *message=None*)

Bases: object

An error object providing more information about the error that occurred.

Parameters

- **object_type** (*(optional)* ask_sdk_model.session_ended_error_type.SessionEndedErrorType) – A string indicating the type of error that occurred.
- **message** (*(optional)* *str*) – A string providing more information about the error.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.session_ended_error_type module

class ask_sdk_model.session_ended_error_type.**SessionEndedErrorType**

Bases: enum.Enum

A string indicating the type of error that occurred.

Allowed enum values: [INVALID_RESPONSE, DEVICE_COMMUNICATION_ERROR, INTERNAL_SERVICE_ERROR]

ask_sdk_model.session_ended_reason module

class ask_sdk_model.session_ended_reason.**SessionEndedReason**

Bases: enum.Enum

The reason why session ended when not initiated from the Skill itself.

Allowed enum values: [USER_INITIATED, ERROR, EXCEEDED_MAX_REPROMPTS]

ask_sdk_model.session_ended_request module

class ask_sdk_model.session_ended_request.**SessionEndedRequest**(*request_id=None*, *timestamp=None*, *locale=None*, *reason=None*, *error=None*)

Bases: *ask_sdk_model.request.Request*

A SessionEndedRequest is an object that represents a request made to an Alexa skill to notify that a session was ended. Your service receives a SessionEndedRequest when a currently open session is closed for one of the following reasons: &lt;ol&gt;&lt;li&gt;The user says “exit”&lt;/li&gt;&lt;li&gt;the user does not respond or says something that does not match an intent defined in your voice interface while the device is listening for the user’s response&lt;/li&gt;&lt;li&gt;an error occurs&lt;/li&gt;&lt;/ol&gt;

Parameters

- **request_id** (*(optional) str*) – Represents the unique identifier for the specific request.
- **timestamp** (*(optional) datetime*) – Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.
- **locale** (*(optional) str*) – A string indicating the user's locale. For example: en-US. This value is only provided with certain request types.
- **reason** (*(optional)* ask_sdk_model.session_ended_reason.SessionEndedReason) – Describes why the session ended.
- **error** (*(optional)* ask_sdk_model.session_ended_error.SessionEndedError) – An error object providing more information about the error that occurred.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.slot module

class ask_sdk_model.slot.**Slot**(*name=None*, *value=None*, *confirmation_status=None*, *resolutions=None*)

Bases: object

Parameters

- **name** (*(optional) str*) – A string that represents the name of the slot.
- **value** (*(optional) str*) – A string that represents the value the user spoke for the slot. This is the actual value the user spoke, not necessarily the canonical value or one of the synonyms defined for the entity. Note that AMAZON.LITERAL slot values sent to your service are always in all lower case.
- **confirmation_status** (*(optional)* ask_sdk_model.slot_confirmation_status.SlotConfirmationStatus) – Indication of whether an intent or slot has been explicitly confirmed or denied by the user, or neither.
- **resolutions** (*(optional)* ask_sdk_model.slu.entityresolution.resolutions.Resolutions) – Contains the results of entity resolution. These are organized by authority. An authority represents the source for the data provided for the slot. For a custom slot type, the authority is the slot type you defined.

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.slot_confirmation_status module

class ask_sdk_model.slot_confirmation_status.**SlotConfirmationStatus**

Bases: enum.Enum

An enumeration indicating whether the user has explicitly confirmed or denied the value of this slot.

Allowed enum values: [NONE, DENIED, CONFIRMED]

ask_sdk_model.supported_interfaces module

class ask_sdk_model.supported_interfaces.**SupportedInterfaces**(*alexa_presentation_apl=None*, *audio_player=None*, *display=None*, *video_app=None*, *geolocation=None*)

Bases: `object`

An object listing each interface that the device supports. For example, if supportedInterfaces includes AudioPlayer {}, then you know that the device supports streaming audio using the AudioPlayer interface.

Parameters

- **alexa_presentation_apl** (*(optional)* ask_sdk_model.interfaces.alexa.presentation.apl.alexa_presentation_apl_interface.AlexaPresentationAplInterface) –
- **audio_player** (*(optional)* ask_sdk_model.interfaces.audioplayer.audio_player_interface.AudioPlayerInterface) –
- **display** (*(optional)* ask_sdk_model.interfaces.display.display_interface.DisplayInterface) –
- **video_app** (*(optional)* ask_sdk_model.interfaces.videoapp.video_app_interface.VideoAppInterface) –
- **geolocation** (*(optional)* ask_sdk_model.interfaces.geolocation.geolocation_interface.GeolocationInterface) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.user module

class ask_sdk_model.user.**User**(*user_id=None*, *access_token=None*, *permissions=None*)

Bases: `object`

Represents the user registered to the device initiating the request.

Parameters

- **user_id** (*(optional)* str) – A string that represents a unique identifier for the user who made the request. The length of this identifier can vary, but is never more than 255 characters. The userId is automatically generated when a user enables the skill in the Alexa app. Note: Disabling and re-enabling a skill generates a new identifier.
- **access_token** (*(optional)* str) – A token identifying the user in another system. This is only provided if the user has successfully linked their account. See Linking an Alexa User with a User in Your System for more details.

- **permissions** (*(optional)* ask_sdk_model.permissions.Permissions) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

ask_sdk_model.permission_status module

class ask_sdk_model.permission_status.**PermissionStatus**

Bases: enum.Enum

This denotes the status of the permission scope.

Allowed enum values: [GRANTED, DENIED]

ask_sdk_model.scope module

class ask_sdk_model.scope.**Scope**(*status=None*)

Bases: object

This is the value of LoginWithAmazon(LWA) consent scope. This object is used as in the key-value pairs that are provided in user.permissions.scopes object

Parameters **status** (*(optional)* ask_sdk_model.permission_status.PermissionStatus) –

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

CHAPTER 3

Got Feedback?

- We would like to hear about your bugs, feature requests, questions or quick feedback. Please search for existing issues before opening a new one. It would also be helpful if you follow the templates for issue and pull request creation. Please follow the contributing guidelines for pull requests!!
- Request and vote for Alexa features!

CHAPTER 4

Additional Resources

4.1 Community

- Amazon Developer Forums : Join the conversation!
- Hackster.io - See what others are building with Alexa.

4.2 Tutorials & Guides

- Voice Design Guide - A great resource for learning conversational and voice user interface design.

CHAPTER 5

Indices and tables

- genindex
- modindex

Python Module Index

Index