

---

# Amazon Lex API Reference



## Amazon Lex API Reference:

Copyright © 2022 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Amazon Lex V2 API Reference .....	1
Actions .....	1
Amazon Lex Model Building V2 .....	3
Amazon Lex Runtime V2 .....	688
Data Types .....	724
Amazon Lex Model Building V2 .....	727
Amazon Lex Runtime V2 .....	891
Common Parameters .....	934
Common Errors .....	936

# Amazon Lex V2 API Reference

This document provides detailed information about the Amazon Lex V2 API actions and their parameters.

For information about the IAM access control permissions you need to use this API, see [Identity-based policies for Amazon Lex](#).

If you use a custom HTTP client to call Amazon Lex Model Building V2 operations, you must set the "Content-Type" HTTP header to "application/x-amz-json-1.1". Otherwise, you receive an HTTP 404 - UnknownOperationException in the response.

Amazon Lex Model Building V2 operations return the responses with the "application/x-amz-json-1.1" content type.

You can use [AWS SDKs](#) to access Amazon Lex V2 APIs using your favorite programming language. The SDKs automatically perform useful tasks for you, such as:

- Cryptographically sign your service requests
- Retry requests
- Handle error responses

The following resources provide additional information about the Amazon Lex V2 API.

- [AWS General Reference](#)
- [Amazon Lex V2 Endpoints for each region](#).
- [AWS Command Line Interface](#)
  - [Amazon Lex Model Building V2 CLI commands](#).
  - [Amazon Lex Runtime V2 CLI commands](#).

## Topics

- [Actions \(p. 1\)](#)
- [Data Types \(p. 724\)](#)
- [Common Parameters \(p. 934\)](#)
- [Common Errors \(p. 936\)](#)

# Actions

The following actions are supported by Amazon Lex Model Building V2:

- [BatchCreateCustomVocabularyItem \(p. 6\)](#)
- [BatchDeleteCustomVocabularyItem \(p. 10\)](#)
- [BatchUpdateCustomVocabularyItem \(p. 14\)](#)
- [BuildBotLocale \(p. 18\)](#)
- [CreateBot \(p. 21\)](#)
- [CreateBotAlias \(p. 26\)](#)
- [CreateBotLocale \(p. 32\)](#)

- [CreateBotVersion \(p. 37\)](#)
- [CreateExport \(p. 41\)](#)
- [CreateIntent \(p. 45\)](#)
- [CreateResourcePolicy \(p. 163\)](#)
- [CreateResourcePolicyStatement \(p. 166\)](#)
- [CreateSlot \(p. 170\)](#)
- [CreateSlotType \(p. 229\)](#)
- [CreateUploadUrl \(p. 235\)](#)
- [DeleteBot \(p. 237\)](#)
- [DeleteBotAlias \(p. 240\)](#)
- [DeleteBotLocale \(p. 243\)](#)
- [DeleteBotVersion \(p. 246\)](#)
- [DeleteCustomVocabulary \(p. 249\)](#)
- [DeleteExport \(p. 252\)](#)
- [DeleteImport \(p. 254\)](#)
- [DeleteIntent \(p. 256\)](#)
- [DeleteResourcePolicy \(p. 259\)](#)
- [DeleteResourcePolicyStatement \(p. 261\)](#)
- [DeleteSlot \(p. 264\)](#)
- [DeleteSlotType \(p. 267\)](#)
- [DeleteUtterances \(p. 270\)](#)
- [DescribeBot \(p. 272\)](#)
- [DescribeBotAlias \(p. 275\)](#)
- [DescribeBotLocale \(p. 279\)](#)
- [DescribeBotRecommendation \(p. 283\)](#)
- [DescribeBotVersion \(p. 287\)](#)
- [DescribeCustomVocabularyMetadata \(p. 291\)](#)
- [DescribeExport \(p. 294\)](#)
- [DescribeImport \(p. 297\)](#)
- [DescribeIntent \(p. 301\)](#)
- [DescribeResourcePolicy \(p. 361\)](#)
- [DescribeSlot \(p. 363\)](#)
- [DescribeSlotType \(p. 394\)](#)
- [ListAggregatedUtterances \(p. 399\)](#)
- [ListBotAliases \(p. 405\)](#)
- [ListBotLocales \(p. 408\)](#)
- [ListBotRecommendations \(p. 412\)](#)
- [ListBots \(p. 416\)](#)
- [ListBotVersions \(p. 419\)](#)
- [ListBuiltInIntents \(p. 422\)](#)
- [ListBuiltInSlotTypes \(p. 425\)](#)
- [ListCustomVocabularyItems \(p. 428\)](#)
- [ListExports \(p. 431\)](#)
- [ListImports \(p. 435\)](#)
- [ListIntents \(p. 439\)](#)
- [ListRecommendedIntents \(p. 443\)](#)

- [ListSlots \(p. 447\)](#)
- [ListSlotTypes \(p. 453\)](#)
- [ListTagsForResource \(p. 457\)](#)
- [SearchAssociatedTranscripts \(p. 459\)](#)
- [StartBotRecommendation \(p. 463\)](#)
- [StartImport \(p. 468\)](#)
- [StopBotRecommendation \(p. 472\)](#)
- [TagResource \(p. 476\)](#)
- [UntagResource \(p. 478\)](#)
- [UpdateBot \(p. 480\)](#)
- [UpdateBotAlias \(p. 484\)](#)
- [UpdateBotLocale \(p. 490\)](#)
- [UpdateBotRecommendation \(p. 495\)](#)
- [UpdateExport \(p. 500\)](#)
- [UpdateIntent \(p. 503\)](#)
- [UpdateResourcePolicy \(p. 621\)](#)
- [UpdateSlot \(p. 624\)](#)
- [UpdateSlotType \(p. 683\)](#)

The following actions are supported by Amazon Lex Runtime V2:

- [DeleteSession \(p. 690\)](#)
- [GetSession \(p. 693\)](#)
- [PutSession \(p. 698\)](#)
- [RecognizeText \(p. 703\)](#)
- [RecognizeUtterance \(p. 710\)](#)
- [StartConversation \(p. 717\)](#)

## Amazon Lex Model Building V2

The following actions are supported by Amazon Lex Model Building V2:

- [BatchCreateCustomVocabularyItem \(p. 6\)](#)
- [BatchDeleteCustomVocabularyItem \(p. 10\)](#)
- [BatchUpdateCustomVocabularyItem \(p. 14\)](#)
- [BuildBotLocale \(p. 18\)](#)
- [CreateBot \(p. 21\)](#)
- [CreateBotAlias \(p. 26\)](#)
- [CreateBotLocale \(p. 32\)](#)
- [CreateBotVersion \(p. 37\)](#)
- [CreateExport \(p. 41\)](#)
- [CreateIntent \(p. 45\)](#)
- [CreateResourcePolicy \(p. 163\)](#)
- [CreateResourcePolicyStatement \(p. 166\)](#)
- [CreateSlot \(p. 170\)](#)
- [CreateSlotType \(p. 229\)](#)

- [CreateUploadUrl \(p. 235\)](#)
- [DeleteBot \(p. 237\)](#)
- [DeleteBotAlias \(p. 240\)](#)
- [DeleteBotLocale \(p. 243\)](#)
- [DeleteBotVersion \(p. 246\)](#)
- [DeleteCustomVocabulary \(p. 249\)](#)
- [DeleteExport \(p. 252\)](#)
- [DeleteImport \(p. 254\)](#)
- [DeleteIntent \(p. 256\)](#)
- [DeleteResourcePolicy \(p. 259\)](#)
- [DeleteResourcePolicyStatement \(p. 261\)](#)
- [DeleteSlot \(p. 264\)](#)
- [DeleteSlotType \(p. 267\)](#)
- [DeleteUtterances \(p. 270\)](#)
- [DescribeBot \(p. 272\)](#)
- [DescribeBotAlias \(p. 275\)](#)
- [DescribeBotLocale \(p. 279\)](#)
- [DescribeBotRecommendation \(p. 283\)](#)
- [DescribeBotVersion \(p. 287\)](#)
- [DescribeCustomVocabularyMetadata \(p. 291\)](#)
- [DescribeExport \(p. 294\)](#)
- [DescribeImport \(p. 297\)](#)
- [DescribeIntent \(p. 301\)](#)
- [DescribeResourcePolicy \(p. 361\)](#)
- [DescribeSlot \(p. 363\)](#)
- [DescribeSlotType \(p. 394\)](#)
- [ListAggregatedUtterances \(p. 399\)](#)
- [ListBotAliases \(p. 405\)](#)
- [ListBotLocales \(p. 408\)](#)
- [ListBotRecommendations \(p. 412\)](#)
- [ListBots \(p. 416\)](#)
- [ListBotVersions \(p. 419\)](#)
- [ListBuiltInIntents \(p. 422\)](#)
- [ListBuiltInSlotTypes \(p. 425\)](#)
- [ListCustomVocabularyItems \(p. 428\)](#)
- [ListExports \(p. 431\)](#)
- [ListImports \(p. 435\)](#)
- [ListIntents \(p. 439\)](#)
- [ListRecommendedIntents \(p. 443\)](#)
- [ListSlots \(p. 447\)](#)
- [ListSlotTypes \(p. 453\)](#)
- [ListTagsForResource \(p. 457\)](#)
- [SearchAssociatedTranscripts \(p. 459\)](#)
- [StartBotRecommendation \(p. 463\)](#)
- [StartImport \(p. 468\)](#)
- [StopBotRecommendation \(p. 472\)](#)

- [TagResource \(p. 476\)](#)
- [UntagResource \(p. 478\)](#)
- [UpdateBot \(p. 480\)](#)
- [UpdateBotAlias \(p. 484\)](#)
- [UpdateBotLocale \(p. 490\)](#)
- [UpdateBotRecommendation \(p. 495\)](#)
- [UpdateExport \(p. 500\)](#)
- [UpdateIntent \(p. 503\)](#)
- [UpdateResourcePolicy \(p. 621\)](#)
- [UpdateSlot \(p. 624\)](#)
- [UpdateSlotType \(p. 683\)](#)

## BatchCreateCustomVocabularyItem

Service: Amazon Lex Model Building V2

Create a batch of custom vocabulary items for a given bot locale's custom vocabulary.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/customvocabulary/DEFAULT/
batchcreate HTTP/1.1
Content-type: application/json

{
    "customVocabularyItemList": [
        {
            "displayAs": "string",
            "phrase": "string",
            "weight": number
        }
    ]
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 6)

The identifier of the bot associated with this custom vocabulary.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 6)

The identifier of the version of the bot associated with this custom vocabulary.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 6)

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see [Supported Languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **customVocabularyItemList** (p. 6)

A list of new custom vocabulary items. Each entry must contain a phrase and can optionally contain a displayAs and/or a weight.

Type: Array of [NewCustomVocabularyItem \(p. 835\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botVersion": "string",
  "errors": [
    {
      "errorCode": "string",
      "errorMessage": "string",
      "itemId": "string"
    }
  ],
  "localeId": "string",
  "resources": [
    {
      "displayAs": "string",
      "itemId": "string",
      "phrase": "string",
      "weight": number
    }
  ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 7\)](#)

The identifier of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 7\)](#)

The identifier of the version of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### [errors \(p. 7\)](#)

A list of custom vocabulary items that failed to create during the operation. The reason for the error is contained within each error object.

Type: Array of [FailedCustomVocabularyItem \(p. 803\)](#) objects

#### **localeId (p. 7)**

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see [Supported Languages](#).

Type: String

#### **resources (p. 7)**

A list of custom vocabulary items that were successfully created during the operation.

Type: Array of [CustomVocabularyItem \(p. 786\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## BatchDeleteCustomVocabularyItem

Service: Amazon Lex Model Building V2

Delete a batch of custom vocabulary items for a given bot locale's custom vocabulary.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/LocaleId/customvocabulary/DEFAULT/
batchdelete HTTP/1.1
Content-type: application/json

{
  "customVocabularyItemList": [
    {
      "itemId": "string"
    }
  ]
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 10)

The identifier of the bot associated with this custom vocabulary.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 10)

The identifier of the version of the bot associated with this custom vocabulary.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 10)

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see [Supported Languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **customVocabularyItemList** (p. 10)

A list of custom vocabulary items requested to be deleted. Each entry must contain the unique custom vocabulary entry identifier.

Type: Array of [CustomVocabularyEntryId](#) (p. 783) objects

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "errors": [
        {
            "errorCode": "string",
            "errorMessage": "string",
            "itemId": "string"
        }
    ],
    "localeId": "string",
    "resources": [
        {
            "displayAs": "string",
            "itemId": "string",
            "phrase": "string",
            "weight": number
        }
    ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId** (p. 11)

The identifier of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion** (p. 11)

The identifier of the version of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### **errors** (p. 11)

A list of custom vocabulary items that failed to delete during the operation. The reason for the error is contained within each error object.

Type: Array of [FailedCustomVocabularyItem](#) (p. 803) objects

### [localeId \(p. 11\)](#)

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see Supported languages (<https://docs.aws.amazon.com/lexv2/latest/dg/how-languages.html>).

Type: String

### [resources \(p. 11\)](#)

A list of custom vocabulary items that were successfully deleted during the operation.

Type: Array of [CustomVocabularyItem \(p. 786\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)

- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## BatchUpdateCustomVocabularyItem

Service: Amazon Lex Model Building V2

Update a batch of custom vocabulary items for a given bot locale's custom vocabulary.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/customvocabulary/DEFAULT/
batchupdate HTTP/1.1
Content-type: application/json

{
  "customVocabularyItemList": [
    {
      "displayAs": "string",
      "itemId": "string",
      "phrase": "string",
      "weight": number
    }
  ]
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 14)

The identifier of the bot associated with this custom vocabulary

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 14)

The identifier of the version of the bot associated with this custom vocabulary.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 14)

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see [Supported Languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **customVocabularyItemList** (p. 14)

A list of custom vocabulary items with updated fields. Each entry must contain a phrase and can optionally contain a displayAs and/or a weight.

Type: Array of [CustomVocabularyItem \(p. 786\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botVersion": "string",
  "errors": [
    {
      "errorCode": "string",
      "errorMessage": "string",
      "itemId": "string"
    }
  ],
  "localeId": "string",
  "resources": [
    {
      "displayAs": "string",
      "itemId": "string",
      "phrase": "string",
      "weight": number
    }
  ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId (p. 15)**

The identifier of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion (p. 15)**

The identifier of the version of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### **errors (p. 15)**

A list of custom vocabulary items that failed to update during the operation. The reason for the error is contained within each error object.

Type: Array of [FailedCustomVocabularyItem \(p. 803\)](#) objects

#### **localeId (p. 15)**

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see [Supported Languages](#).

Type: String

#### **resources (p. 15)**

A list of custom vocabulary items that were successfully updated during the operation.

Type: Array of [CustomVocabularyItem \(p. 786\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## BuildBotLocale

Service: Amazon Lex Model Building V2

Builds a bot, its intents, and its slot types into a specific locale. A bot can be built into multiple locales. At runtime the locale is used to choose a specific build of the bot.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 18)

The identifier of the bot to build. The identifier is returned in the response from the [CreateBot](#) operation.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 18)

The version of the bot to build. This can only be the draft version of the bot.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 18)

The identifier of the language and locale that the bot will be used in. The string must match one of the supported locales. All of the intents, slot types, and slots used in the bot must have the same locale. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botId": "string",
    "botLocaleStatus": "string",
    "botVersion": "string",
    "lastBuildSubmittedDateTime": number,
    "localeId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### **botId (p. 18)**

The identifier of the specified bot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botLocaleStatus (p. 18)**

The bot's build status. When the status is ReadyExpressTesting you can test the bot using the utterances defined for the intents and slot types. When the status is Built, the bot is ready for use and can be tested using any utterance.

Type: String

Valid Values: Creating | Building | Built | ReadyExpressTesting | Failed | Deleting | NotBuilt | Importing | Processing

### **botVersion (p. 18)**

The version of the bot that was built. This is only the draft version of the bot.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **lastBuildSubmittedDateTime (p. 18)**

A timestamp indicating the date and time that the bot was last built for this locale.

Type: Timestamp

### **localeId (p. 18)**

The language and locale specified of where the bot can be used.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateBot

Service: Amazon Lex Model Building V2

Creates an Amazon Lex conversational bot.

### Request Syntax

```
PUT /bots/ HTTP/1.1
Content-type: application/json

{
    "botName": "string",
    "botTags": {
        "string" : "string"
    },
    "dataPrivacy": {
        "childDirected": boolean
    },
    "description": "string",
    "idleSessionTTLInSeconds": number,
    "roleArn": "string",
    "testBotAliasTags": {
        "string" : "string"
    }
}
```

### URI Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### **botName (p. 21)**

The name of the bot. The bot name must be unique in the account that creates the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### **botTags (p. 21)**

A list of tags to add to the bot. You can only add tags when you create a bot. You can't use the UpdateBot operation to update tags. To update tags, use the TagResource operation.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

### [dataPrivacy \(p. 21\)](#)

Provides information on additional privacy protections Amazon Lex should use with the bot's data.

Type: [DataPrivacy \(p. 787\)](#) object

Required: Yes

### [description \(p. 21\)](#)

A description of the bot. It appears in lists to help you identify a particular bot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

### [idleSessionTTLInSeconds \(p. 21\)](#)

The time, in seconds, that Amazon Lex should keep information about a user's conversation with the bot.

A user interaction remains active for the amount of time specified. If no conversation occurs during this time, the session expires and Amazon Lex deletes any data provided before the timeout.

You can specify between 60 (1 minute) and 86,400 (24 hours) seconds.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 86400.

Required: Yes

### [roleArn \(p. 21\)](#)

The Amazon Resource Name (ARN) of an IAM role that has permission to access the bot.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:iam::[0-9]{12}:role/.\*\$

Required: Yes

### [testBotAliasTags \(p. 21\)](#)

A list of tags to add to the test alias for a bot. You can only add tags when you create a bot. You can't use the UpdateAlias operation to update tags. To update tags on the test alias, use the TagResource operation.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

## Response Syntax

HTTP/1.1 202
--------------

```
Content-type: application/json

{
    "botId": "string",
    "botName": "string",
    "botStatus": "string",
    "botTags": {
        "string" : "string"
    },
    "creationDateTime": number,
    "dataPrivacy": {
        "childDirected": boolean
    },
    "description": "string",
    "idleSessionTTLInSeconds": number,
    "roleArn": "string",
    "testBotAliasTags": {
        "string" : "string"
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### **botId** (p. 22)

A unique identifier for a particular bot. You use this to identify the bot when you call other Amazon Lex API operations.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botName** (p. 22)

The name specified for the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

### **botStatus** (p. 22)

Shows the current status of the bot. The bot is first in the Creating status. Once the bot is ready for use, it changes to the Available status. After the bot is created, you can use the Draft version of the bot.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

### **botTags** (p. 22)

A list of tags associated with the bot.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

#### **creationDateTime (p. 22)**

A timestamp indicating the date and time that the bot was created.

Type: Timestamp

#### **dataPrivacy (p. 22)**

The data privacy settings specified for the bot.

Type: [DataPrivacy \(p. 787\)](#) object

#### **description (p. 22)**

The description specified for the bot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

#### **idleSessionTTLInSeconds (p. 22)**

The session idle time specified for the bot.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 86400.

#### **roleArn (p. 22)**

The IAM role specified for the bot.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:iam::[0-9]{12}:role/.+\$

#### **testBotAliasTags (p. 22)**

A list of tags associated with the test alias for the bot.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

## **Errors**

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

#### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

**InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateBotAlias

Service: Amazon Lex Model Building V2

Creates an alias for the specified version of a bot. Use an alias to enable you to change the version of a bot without updating applications that use the bot.

For example, you can create an alias called "PROD" that your applications use to call the Amazon Lex bot.

### Request Syntax

```
PUT /bots/botId/botaliases/ HTTP/1.1
Content-type: application/json

{
    "botAliasLocaleSettings": {
        "string" : {
            "codeHookSpecification": {
                "lambdaCodeHook": {
                    "codeHookInterfaceVersion": "string",
                    "lambdaARN": "string"
                }
            },
            "enabled": boolean
        }
    },
    "botAliasName": "string",
    "botVersion": "string",
    "conversationLogSettings": {
        "audioLogSettings": [
            {
                "destination": {
                    "s3Bucket": {
                        "kmsKeyArn": "string",
                        "logPrefix": "string",
                        "s3BucketArn": "string"
                    }
                },
                "enabled": boolean
            }
        ],
        "textLogSettings": [
            {
                "destination": {
                    "cloudWatch": {
                        "cloudWatchLogGroupArn": "string",
                        "logPrefix": "string"
                    }
                },
                "enabled": boolean
            }
        ]
    },
    "description": "string",
    "sentimentAnalysisSettings": {
        "detectSentiment": boolean
    },
    "tags": {
        "string": "string"
    }
}
```

## URI Request Parameters

The request uses the following URI parameters.

### **botId (p. 26)**

The unique identifier of the bot that the alias applies to.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### **botAliasLocaleSettings (p. 26)**

Maps configuration information to a specific locale. You can use this parameter to specify a specific Lambda function to run different functions in different locales.

Type: String to [BotAliasLocaleSettings \(p. 745\)](#) object map

Map Entries: Maximum number of items.

Required: No

### **botAliasName (p. 26)**

The alias to create. The name must be unique for the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

### **botVersion (p. 26)**

The version of the bot that this alias points to. You can use the [UpdateBotAlias](#) operation to change the bot version associated with the alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

Required: No

### **conversationLogSettings (p. 26)**

Specifies whether Amazon Lex logs text and audio for a conversation with the bot. When you enable conversation logs, text logs store text input, transcripts of audio input, and associated metadata in Amazon CloudWatch Logs. Audio logs store audio input in Amazon S3.

Type: [ConversationLogSettings \(p. 781\)](#) object

Required: No

#### [description \(p. 26\)](#)

A description of the alias. Use this description to help identify the alias.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### [sentimentAnalysisSettings \(p. 26\)](#)

Determines whether Amazon Lex will use Amazon Comprehend to detect the sentiment of user utterances.

Type: [SentimentAnalysisSettings \(p. 856\)](#) object

Required: No

#### [tags \(p. 26\)](#)

A list of tags to add to the bot alias. You can only add tags when you create an alias, you can't use the UpdateBotAlias operation to update the tags on a bot alias. To update tags, use the TagResource operation.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botAliasId": "string",
    "botAliasLocaleSettings": {
        "string" : {
            "codeHookSpecification": {
                "lambdaCodeHook": {
                    "codeHookInterfaceVersion": "string",
                    "lambdaARN": "string"
                }
            },
            "enabled": boolean
        }
    },
    "botAliasName": "string",
    "botAliasStatus": "string",
    "botId": "string",
    "botVersion": "string",
    "conversationLogSettings": {
        "audioLogSettings": [
            {
                "destination": {

```

```
        "s3Bucket": {
            "kmsKeyArn": "string",
            "logPrefix": "string",
            "s3BucketArn": "string"
        }
    },
    "enabled": boolean
}
],
"textLogSettings": [
{
    "destination": {
        "cloudWatch": {
            "cloudWatchLogGroupArn": "string",
            "logPrefix": "string"
        }
    },
    "enabled": boolean
}
]
},
"creationDateTime": number,
"description": string,
"sentimentAnalysisSettings": {
    "detectSentiment": boolean
},
"tags": {
    "string" : "string"
}
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### [botAliasId \(p. 28\)](#)

The unique identifier of the bot alias.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

### [botAliasLocaleSettings \(p. 28\)](#)

Configuration information for a specific locale.

Type: String to [BotAliasLocaleSettings \(p. 745\)](#) object map

Map Entries: Maximum number of items.

### [botAliasName \(p. 28\)](#)

The name specified for the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

**botAliasStatus (p. 28)**

The current status of the alias. The alias is first put into the Creating state. When the alias is ready to be used, it is put into the Available state. You can use the `DescribeBotAlias` operation to get the current state of an alias.

Type: String

Valid Values: Creating | Available | Deleting | Failed

**botId (p. 28)**

The unique identifier of the bot that this alias applies to.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botVersion (p. 28)**

The version of the bot associated with this alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

**conversationLogSettings (p. 28)**

The conversation log settings specified for the alias.

Type: [ConversationLogSettings \(p. 781\)](#) object

**creationDateTime (p. 28)**

A Unix timestamp indicating the date and time that the bot alias was created.

Type: Timestamp

**description (p. 28)**

The description specified for the bot alias.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**sentimentAnalysisSettings (p. 28)**

Determines whether Amazon Lex will use Amazon Comprehend to detect the sentiment of user utterances.

Type: [SentimentAnalysisSettings \(p. 856\)](#) object

**tags (p. 28)**

A list of tags associated with the bot alias.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateBotLocale

Service: Amazon Lex Model Building V2

Creates a locale in the bot. The locale contains the intents and slot types that the bot uses in conversations with users in the specified language and locale. You must add a locale to a bot before you can add intents and slot types to the bot.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/ HTTP/1.1
Content-type: application/json

{
  "description": "string",
  "localeId": "string",
  "nluIntentConfidenceThreshold": number,
  "voiceSettings": {
    "engine": "string",
    "voiceId": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 32)

The identifier of the bot to create the locale for.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 32)

The version of the bot to create the locale for. This can only be the draft version of the bot.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **description** (p. 32)

A description of the bot locale. Use this to help identify the bot locale in lists.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

### [localeId \(p. 32\)](#)

The identifier of the language and locale that the bot will be used in. The string must match one of the supported locales. All of the intents, slot types, and slots used in the bot must have the same locale. For more information, see [Supported languages](#).

Type: String

Required: Yes

### [nluIntentConfidenceThreshold \(p. 32\)](#)

Determines the threshold where Amazon Lex will insert the AMAZON.FallbackIntent, AMAZON.KendraSearchIntent, or both when returning alternative intents. AMAZON.FallbackIntent and AMAZON.KendraSearchIntent are only inserted if they are configured for the bot.

For example, suppose a bot is configured with the confidence threshold of 0.80 and the AMAZON.FallbackIntent. Amazon Lex returns three alternative intents with the following confidence scores: IntentA (0.70), IntentB (0.60), IntentC (0.50). The response from the RecognizeText operation would be:

- AMAZON.FallbackIntent
- IntentA
- IntentB
- IntentC

Type: Double

Valid Range: Minimum value of 0. Maximum value of 1.

Required: Yes

### [voiceSettings \(p. 32\)](#)

The Amazon Polly voice ID that Amazon Lex uses for voice interaction with the user.

Type: [VoiceSettings \(p. 890\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "botId": "string",
  "botLocaleStatus": "string",
  "botVersion": "string",
  "creationDateTime": number,
  "description": "string",
  "localeId": "string",
  "localeName": "string",
  "nluIntentConfidenceThreshold": number,
  "voiceSettings": {
    "engine": "string",
    "voiceId": "string"
  }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### **botId (p. 33)**

The specified bot identifier.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botLocaleStatus (p. 33)**

The status of the bot.

When the status is `Creating` the bot locale is being configured. When the status is `Building` Amazon Lex is building the bot for testing and use.

If the status of the bot is `ReadyExpressTesting`, you can test the bot using the exact utterances specified in the bots' intents. When the bot is ready for full testing or to run, the status is `Built`.

If there was a problem with building the bot, the status is `Failed`. If the bot was saved but not built, the status is `NotBuilt`.

Type: String

Valid Values: `Creating` | `Building` | `Built` | `ReadyExpressTesting` | `Failed` | `Deleting` | `NotBuilt` | `Importing` | `Processing`

### **botVersion (p. 33)**

The specified bot version.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **creationDateTime (p. 33)**

A timestamp specifying the date and time that the bot locale was created.

Type: Timestamp

### **description (p. 33)**

The specified description of the bot locale.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

### **localeId (p. 33)**

The specified locale identifier.

Type: String

### **localeName (p. 33)**

The specified locale name.

Type: String

[nluIntentConfidenceThreshold \(p. 33\)](#)

The specified confidence threshold for inserting the AMAZON.FallbackIntent and AMAZON.KendraSearchIntent intents.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 1.

[voiceSettings \(p. 33\)](#)

The Amazon Polly voice ID that Amazon Lex uses for voice interaction with the user.

Type: [VoiceSettings \(p. 890\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

**ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

**InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateBotVersion

Service: Amazon Lex Model Building V2

Creates a new version of the bot based on the DRAFT version. If the DRAFT version of this resource hasn't changed since you created the last version, Amazon Lex doesn't create a new version, it returns the last created version.

When you create the first version of a bot, Amazon Lex sets the version to 1. Subsequent versions increment by 1.

### Request Syntax

```
PUT /bots/botId/botversions/ HTTP/1.1
Content-type: application/json

{
  "botVersionLocaleSpecification": {
    "string": {
      "sourceBotVersion": "string"
    }
  },
  "description": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId (p. 37)**

The identifier of the bot to create the version for.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **botVersionLocaleSpecification (p. 37)**

Specifies the locales that Amazon Lex adds to this version. You can choose the Draft version or any other previously published version for each locale. When you specify a source version, the locale data is copied from the source version to the new version.

Type: String to [BotVersionLocaleDetails \(p. 766\)](#) object map

Map Entries: Maximum number of items.

Required: Yes

#### **description (p. 37)**

A description of the version. Use the description to help identify the version in lists.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botId": "string",
    "botStatus": "string",
    "botVersion": "string",
    "botVersionLocaleSpecification": [
        "string" : {
            "sourceBotVersion": "string"
        }
    ],
    "creationDateTime": number,
    "description": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### **botId** (p. 38)

The bot identifier specified in the request.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botStatus** (p. 38)

When you send a request to create or update a bot, Amazon Lex sets the status response element to Creating. After Amazon Lex builds the bot, it sets status to Available. If Amazon Lex can't build the bot, it sets status to Failed.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

### **botVersion** (p. 38)

The version number assigned to the version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

### **botVersionLocaleSpecification** (p. 38)

The source versions used for each locale in the new version.

Type: String to [BotVersionLocaleDetails \(p. 766\)](#) object map

Map Entries: Maximum number of items.

#### [creationDateTime \(p. 38\)](#)

A timestamp of the date and time that the version was created.

Type: Timestamp

#### [description \(p. 38\)](#)

The description of the version specified in the request.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateExport

Service: Amazon Lex Model Building V2

Creates a zip archive containing the contents of a bot or a bot locale. The archive contains a directory structure that contains JSON files that define the bot.

You can create an archive that contains the complete definition of a bot, or you can specify that the archive contain only the definition of a single bot locale.

For more information about exporting bots, and about the structure of the export archive, see [Importing and exporting bots](#)

### Request Syntax

```
PUT /exports/ HTTP/1.1
Content-type: application/json

{
  "fileFormat": "string",
  "filePassword": "string",
  "resourceSpecification": {
    "botExportSpecification": {
      "botId": "string",
      "botVersion": "string"
    },
    "botLocaleExportSpecification": {
      "botId": "string",
      "botVersion": "string",
      "localeId": "string"
    },
    "customVocabularyExportSpecification": {
      "botId": "string",
      "botVersion": "string",
      "localeId": "string"
    }
  }
}
```

### URI Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### fileFormat (p. 41)

The file format of the bot or bot locale definition files.

Type: String

Valid Values: LexJson | TSV

Required: Yes

#### filePassword (p. 41)

An password to use to encrypt the exported archive. Using a password is optional, but you should encrypt the archive to protect the data in transit between Amazon Lex and your local computer.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

#### [resourceSpecification \(p. 41\)](#)

Specifies the type of resource to export, either a bot or a bot locale. You can only specify one type of resource to export.

Type: [ExportResourceSpecification \(p. 798\)](#) object

Required: Yes

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "creationDateTime": number,
  "exportId": "string",
  "exportStatus": "string",
  "fileFormat": "string",
  "resourceSpecification": {
    "botExportSpecification": {
      "botId": "string",
      "botVersion": "string"
    },
    "botLocaleExportSpecification": {
      "botId": "string",
      "botVersion": "string",
      "localeId": "string"
    },
    "customVocabularyExportSpecification": {
      "botId": "string",
      "botVersion": "string",
      "localeId": "string"
    }
  }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

#### [creationDateTime \(p. 42\)](#)

The date and time that the request to export a bot was created.

Type: Timestamp

#### [exportId \(p. 42\)](#)

An identifier for a specific request to create an export.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [exportStatus \(p. 42\)](#)

The status of the export. When the status is Completed, you can use the [DescribeExport](#) operation to get the pre-signed S3 URL link to your exported bot or bot locale.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

### [fileFormat \(p. 42\)](#)

The file format used for the bot or bot locale definition files.

Type: String

Valid Values: LexJson | TSV

### [resourceSpecification \(p. 42\)](#)

A description of the type of resource that was exported, either a bot or a bot locale.

Type: [ExportResourceSpecification \(p. 798\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateIntent

Service: Amazon Lex Model Building V2

Creates an intent.

To define the interaction between the user and your bot, you define one or more intents. For example, for a pizza ordering bot you would create an OrderPizza intent.

When you create an intent, you must provide a name. You can optionally provide the following:

- Sample utterances. For example, "I want to order a pizza" and "Can I order a pizza." You can't provide utterances for built-in intents.
- Information to be gathered. You specify slots for the information that your bot requests from the user. You can specify standard slot types, such as date and time, or custom slot types for your application.
- How the intent is fulfilled. You can provide a Lambda function or configure the intent to return the intent information to your client application. If you use a Lambda function, Amazon Lex invokes the function when all of the intent information is available.
- A confirmation prompt to send to the user to confirm an intent. For example, "Shall I order your pizza?"
- A conclusion statement to send to the user after the intent is fulfilled. For example, "I ordered your pizza."
- A follow-up prompt that asks the user for additional activity. For example, "Do you want a drink with your pizza?"

## Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/intents/ HTTP/1.1
Content-type: application/json
```

```
{
  "description": "string",
  "dialogCodeHook": {
    "enabled": boolean
  },
  "fulfillmentCodeHook": {
    "active": boolean,
    "enabled": boolean,
    "fulfillmentUpdatesSpecification": {
      "active": boolean,
      "startResponse": {
        "allowInterrupt": boolean,
        "delayInSeconds": number,
        "messageGroups": [
          {
            "message": {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              }
            }
          }
        ]
      }
    }
  }
}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"timeoutInSeconds": number,
"updateResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
},
"postFulfillmentStatusSpecification": {
    "failureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                        "type": "string"
                    },
                    "intent": {
                        "name": "string",
                        "slots": {
                            "string" : {
                                "shape": "string",
                                "value": {
                                    "interpretedValue": "string"
                                },
                                "values": [
                                    "SlotValueOverride"
                                ]
                            }
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            }
        ],
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {

```

```
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  }
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    }
  }
}
```

```
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    },
    "failureNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        }
    }
}
```

```
"intent": {  
    "name": "string",  
    "slots": {  
        "string" : {  
            "shape": "string",  
            "value": {  
                "interpretedValue": "string"  
            },  
            "values": [  
                "SlotValueOverride"  
            ]  
        }  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"failureResponse": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    },  
                    "ssmlMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        }
      ],
    },
  ],
  "successConditional": {
    "active": boolean,
    "conditionalBranches": [
      {
        "condition": {
          "expressionString": "string"
        },
        "name": "string",
        "nextStep": {
          "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
          },
          "intent": {
            "name": "string",
            "slots": {
              "string": {
                "shape": "string",
                "value": {
                  "interpretedValue": "string"
                },
                "values": [
                  "SlotValueOverride"
                ]
              }
            }
          }
        },
        "sessionAttributes": {
          "string": "string"
        }
      },
      "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
          {
            "message": {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              },
              "plainTextMessage": {
                "value": "string"
              },
              "ssmlMessage": {
                "value": "string"
              }
            },
            "variations": [
              {
                "customPayload": {

```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ]
                }
            }
        }
    ]
}
```

```
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
```

```
"messageGroups": [
  {
    "message": {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  }
},
"timeoutConditional": {
  "active": boolean,
  "conditionalBranches": [
    {
      "condition": {
        "expressionString": "string"
      },
      "name": "string",
      "nextStep": {
        "dialogAction": {
          "slotToElicit": "string",
          "suppressNextMessage": boolean,
          "type": "string"
        },
        "intent": {

```

```
"name": "string",
"slots": {
    "string" : {
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ]
      }
    ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": [
              "interpretedValue": "string"
            ],
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              }
            }
          ]
        }
      ]
    }
  }
}
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
}
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
            }
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"initialResponseSetting": {
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {
                "active": boolean,
                "conditionalBranches": [
                    {
                        "condition": {
                            "expressionString": "string"
                        },
                        "name": "string",
                        "nextStep": {
                            "dialogAction": {
                                "slotToElicit": "string",
                                "suppressNextMessage": boolean,
                                "type": "string"
                            },
                            "intent": {
                                "name": "string",
                                "slots": {
                                    "string" : {
                                        "shape": "string",
                                        "value": {
                                            "interpretedValue": "string"
                                        },
                                        "values": [
                                            "SlotValueOverride"
                                        ]
                                    }
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

```
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ],
        "defaultBranch": {
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                }
            }
        }
    }
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
}
```

```
        }
      ]
    }
  ],
  "failureNextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {

```

```
        "text": "string",
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        }
    ],
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "image": "string"
                    }
                }
            }
        ]
    }
}
```

```
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
```

```
"allowInterrupt": boolean,
"messageGroups": [
  {
    "message": {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  }
},
"successNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string": {
        "shape": "string",
        "value": {
          "string": "string"
        }
      }
    }
  }
}
```

```
        "interpretedValue": "string"
    },
    "values": [
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "timeoutConditional": {

```

```
"active": boolean,
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
]
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {
                "value": "string"
            }
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "text": "string"
            }
        }
    ]
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"conditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": "string"
                        }
                    }
                }
            }
        }
    ]
}
}
```

```
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
    ],
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {
                "name": "string",
                "slots": {
                    "string" : {
                        "shape": "string",
                        "value": {
                            "interpretedValue": "string"
                        },
                        "values": [
                            "SlotValueOverride"
                        ]
                    }
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"initialResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
    ],
},
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"inputContexts": [
{
    "name": "string"
}
],
"intentClosingSetting": {
    "active": boolean,
    "closingResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },

```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"conditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        }
    ],
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ]
                    }
                }
            }
        ]
    }
}
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
}
```

```
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ]
    },
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
```

```
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
}
},
"intentConfirmationSetting": {
    "active": boolean,
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {
                "active": boolean,
                "conditionalBranches": [
                    {
                        "condition": {
                            "expressionString": "string"
                        },
                        "name": "string",
                        "nextStep": {
                            "dialogAction": {
                                "slotToElicit": "string",
                                "suppressNextMessage": boolean,
                                "type": "string"
                            },
                            "intent": {
                                "name": "string",
                                "slots": {
                                    "string" : {
                                        "shape": "string",
                                        "value": {
                                            "interpretedValue": "string"
                                        },
                                        "values": [
                                            "SlotValueOverride"
                                        ]
                                    }
                                }
                            }
                        },
                        "sessionAttributes": {
                            "string" : "string"
                        }
                    }
                ]
            }
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "string" : "string"
                        }
                    ]
                }
            }
        }
    ]
}
```

```
        "text": "string",
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string": {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    },
    "sessionAttributes": {
```

```
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "failureNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",

```

```
    "slots": {
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
    ],
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": "string"
                            },
                            "text": "string"
                        }
                    ]
                }
            ]
        }
    ]
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                }
            }
        ]
    }
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
    {

```

```
"message": {
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {

```

```
"string" : {
    "shape": "string",
    "value": {
        "interpretedValue": "string"
    },
    "values": [
        "SlotValueOverride"
    ]
}
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ]
    }
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ]
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"confirmationConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [

```

```
{  
    "message": {  
        "customPayload": {  
            "value": "string"  
        },  
        "imageResponseCard": {  
            "buttons": [  
                {  
                    "text": "string",  
                    "value": "string"  
                }  
            ],  
            "imageUrl": "string",  
            "subtitle": "string",  
            "title": "string"  
        },  
        "plainTextMessage": {  
            "value": "string"  
        },  
        "ssmlMessage": {  
            "value": "string"  
        }  
    },  
    "variations": [  
        {  
            "customPayload": {  
                "value": "string"  
            },  
            "imageResponseCard": {  
                "buttons": [  
                    {  
                        "text": "string",  
                        "value": "string"  
                    }  
                ],  
                "imageUrl": "string",  
                "subtitle": "string",  
                "title": "string"  
            },  
            "plainTextMessage": {  
                "value": "string"  
            },  
            "ssmlMessage": {  
                "value": "string"  
            }  
        }  
    ]  
},  
"defaultBranch": {  
    "nextStep": {  
        "dialogAction": {  
            "slotToElicit": "string",  
            "suppressNextMessage": boolean,  
            "type": "string"  
        },  
        "intent": {  
            "name": "string",  
            "slots": {  
                "string" : {  
                    "shape": "string",  
                    "value": {  
                        "interpretedValue": "string"  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
```

```
"confirmationNextStep": {  
    "dialogAction": {  
        "slotToElicit": "string",  
        "suppressNextMessage": boolean,  
        "type": "string"  
    },  
    "intent": {  
        "name": "string",  
        "slots": {  
            "string" : {  
                "shape": "string",  
                "value": {  
                    "interpretedValue": "string"  
                },  
                "values": [  
                    "SlotValueOverride"  
                ]  
            }  
        }  
    },  
    "sessionAttributes": {  
        "string" : "string"  
    }  
},  
"confirmationResponse": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "slotToElicit": "string"  
                },  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "slotToElicit": "string"  
                }  
            ]  
        }  
    ]  
}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
},
"declinationConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string": {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string": "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {

```

```
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"declinationNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
},
```

```
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "declinationResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ],
        "elicitationCodeHook": {
            "enableCodeHookInvocation": boolean,
            "invocationLabel": "string"
        },
        "failureConditional": {
            "active": boolean,
            "conditionalBranches": [
                {
                    "condition": {

```

```
        "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "title": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        ]
    }
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"promptSpecification": {
    "allowInterrupt": boolean,
    "maxRetries": number,
    "messageGroups": [
{
    "message": {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
]
}
```

```
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeTimeoutMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeTimeoutMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
}
},
"intentName": "string",
"kendraConfiguration": {
    "kendraIndex": "string",
    "queryFilterString": "string",
    "queryFilterStringEnabled": boolean
},
"outputContexts": [
    {
        "name": "string",
        "timeToLiveInSeconds": number,
        "value": "string"
    }
]
```

```
        "turnsToLive": number
    }
],
"parentIntentSignature": "string",
"sampleUtterances": [
{
    "utterance": "string"
}
]
}
```

## URI Request Parameters

The request uses the following URI parameters.

### **botId** (p. 45)

The identifier of the bot associated with this intent.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **botVersion** (p. 45)

The identifier of the version of the bot associated with this intent.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

### **localeId** (p. 45)

The identifier of the language and locale where this intent is used. All of the bots, slot types, and slots used by the intent must have the same locale. For more information, see [Supported languages](#).

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### **description** (p. 45)

A description of the intent. Use the description to help identify the intent in lists.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

### **dialogCodeHook** (p. 45)

Specifies that Amazon Lex invokes the alias Lambda function for each user input. You can invoke this Lambda function to personalize user interaction.

For example, suppose that your bot determines that the user's name is John. You Lambda function might retrieve John's information from a backend database and prepopulate some of the values.

For example, if you find that John is gluten intolerant, you might set the corresponding intent slot, `glutenIntolerant` to `true`. You might find John's phone number and set the corresponding session attribute.

Type: [DialogCodeHookSettings \(p. 792\)](#) object

Required: No

#### [fulfillmentCodeHook \(p. 45\)](#)

Specifies that Amazon Lex invokes the alias Lambda function when the intent is ready for fulfillment. You can invoke this function to complete the bot's transaction with the user.

For example, in a pizza ordering bot, the Lambda function can look up the closest pizza restaurant to the customer's location and then place an order on the customer's behalf.

Type: [FulfillmentCodeHookSettings \(p. 804\)](#) object

Required: No

#### [initialResponseSetting \(p. 45\)](#)

Configuration settings for the response that is sent to the user at the beginning of a conversation, before eliciting slot values.

Type: [InitialResponseSetting \(p. 817\)](#) object

Required: No

#### [inputContexts \(p. 45\)](#)

A list of contexts that must be active for this intent to be considered by Amazon Lex.

When an intent has an input context list, Amazon Lex only considers using the intent in an interaction with the user when the specified contexts are included in the active context list for the session. If the contexts are not active, then Amazon Lex will not use the intent.

A context can be automatically activated using the `outputContexts` property or it can be set at runtime.

For example, if there are two intents with different input contexts that respond to the same utterances, only the intent with the active context will respond.

An intent may have up to 5 input contexts. If an intent has multiple input contexts, all of the contexts must be active to consider the intent.

Type: Array of [InputContext \(p. 818\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

Required: No

#### [intentClosingSetting \(p. 45\)](#)

Sets the response that Amazon Lex sends to the user when the intent is closed.

Type: [IntentClosingSetting \(p. 819\)](#) object

Required: No

#### [intentConfirmationSetting \(p. 45\)](#)

Provides prompts that Amazon Lex sends to the user to confirm the completion of an intent. If the user answers "no," the settings contain a statement that is sent to the user to end the intent.

Type: [IntentConfirmationSetting \(p. 820\)](#) object

Required: No

#### [intentName \(p. 45\)](#)

The name of the intent. Intent names must be unique in the locale that contains the intent and cannot match the name of any built-in intent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### [kendraConfiguration \(p. 45\)](#)

Configuration information required to use the AMAZON.KendraSearchIntent intent to connect to an Amazon Kendra index. The AMAZON.KendraSearchIntent intent is called when Amazon Lex can't determine another intent to invoke.

Type: [KendraConfiguration \(p. 829\)](#) object

Required: No

#### [outputContexts \(p. 45\)](#)

A lists of contexts that the intent activates when it is fulfilled.

You can use an output context to indicate the intents that Amazon Lex should consider for the next turn of the conversation with a customer.

When you use the `outputContextsList` property, all of the contexts specified in the list are activated when the intent is fulfilled. You can set up to 10 output contexts. You can also set the number of conversation turns that the context should be active, or the length of time that the context should be active.

Type: Array of [OutputContext \(p. 837\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

#### [parentIntentSignature \(p. 45\)](#)

A unique identifier for the built-in intent to base this intent on.

Type: String

Required: No

#### [sampleUtterances \(p. 45\)](#)

An array of strings that a user might say to signal the intent. For example, "I want a pizza", or "I want a {PizzaSize} pizza".

In an utterance, slot names are enclosed in curly braces ("{" , "}") to indicate where they should be displayed in the utterance shown to the user..

Type: Array of [SampleUtterance \(p. 854\)](#) objects

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "creationDateTime": number,
    "description": "string",
    "dialogCodeHook": {
        "enabled": boolean
    },
    "fulfillmentCodeHook": {
        "active": boolean,
        "enabled": boolean,
        "fulfillmentUpdatesSpecification": {
            "active": boolean,
            "startResponse": {
                "allowInterrupt": boolean,
                "delayInSeconds": number,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "imageUrl": "string",
                                "subtitle": "string",
                                "title": "string"
                            }
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "imageUrl": "string",
                                "subtitle": "string",
                                "title": "string"
                            }
                        },
                        "plainTextMessage": {

```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"timeoutInSeconds": number,
"updateResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        ]
    }
},
]
},
```

```
"postFulfillmentStatusSpecification": {
    "failureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                        "type": "string"
                    },
                    "intent": {
                        "name": "string",
                        "slots": {
                            "string": {
                                "shape": "string",
                                "value": {
                                    "interpretedValue": "string"
                                }
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string": "string"
            }
        ],
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [

```

```
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  }
},
"response": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        }
      }
    }
  ]
}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": "string"
                        }
                    }
                }
            }
        }
    ]
}
```

```
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
    ],
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {
                "name": "string",
                "slots": {
                    "string" : {
                        "shape": "string",
                        "value": {
                            "interpretedValue": "string"
                        },
                        "values": [
                            "SlotValueOverride"
                        ]
                    }
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
],
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string": {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string": "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        }
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    }
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
    }
}
```

```
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {

```

```
        "value": "string"
    }
}
]
}
},
"initialResponseSetting": {
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": string,
        "postCodeHookSpecification": {
            "failureConditional": {
                "active": boolean,
                "conditionalBranches": [
                    {
                        "condition": {
                            "expressionString": "string"
                        },
                        "name": string,
                        "nextStep": {
                            "dialogAction": {
                                "slotToElicit": "string",
                                "suppressNextMessage": boolean,
                                "type": "string"
                            },
                            "intent": {
                                "name": "string",
                                "slots": {
                                    "string" : {
                                        "shape": "string",
                                        "value": {
                                            "interpretedValue": "string"
                                        },
                                        "values": [
                                            "SlotValueOverride"
                                        ]
                                    }
                                }
                            },
                            "sessionAttributes": {
                                "string" : "string"
                            }
                        }
                    }
                ],
                "response": {
                    "allowInterrupt": boolean,
                    "messageGroups": [
                        {
                            "message": {
                                "customPayload": {
                                    "value": "string"
                                },
                                "imageResponseCard": {
                                    "buttons": [
                                        {
                                            "text": "string",
                                            "value": "string"
                                        }
                                    ],
                                    "imageUrl": "string",
                                    "subtitle": "string",
                                    "title": "string"
                                }
                            }
                        }
                    ]
                }
            }
        }
    }
}
```

```
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {

```

```
    "customPayload": {
      "value": "string"
    },
    "imageResponseCard": {
      "buttons": [
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  },
  "variations": [
    {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    }
  ]
}
},
"failureNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        }
      },
      "values": [
        "SlotValueOverride"
      ]
    }
  }
}
```

```
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "successConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {

```

```
        "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        ]
    }
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            }
        }
    ]
}
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ],
        "defaultBranch": {
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",

```

```
    "suppressNextMessage": boolean,
    "type": "string"
},
"intent": {
    "name": "string",
    "slots": {
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
}
},
"conditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {

```

```
        "text": "string",
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {

```

```
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        ]
    }
},
"initialResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
}
```

```
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
```

```
        "string" : "string"
    }
},
"inputContexts": [
{
  "name": "string"
}
],
"intentClosingSetting": {
  "active": boolean,
  "closingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
    {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    }
  ]
}
],
"conditional": {
  "active": boolean,
```

```
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ]
            }
          }
        ]
      }
    ]
  }
]
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"intentConfirmationSetting": {
    "active": boolean,
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {

```

```
"active": boolean,
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
]
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {
                "value": "string"
            }
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "text": "string"
            }
        }
    ]
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": [
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            }
                        }
                    }
                }
            }
        }
    ]
}
```

```
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
```

```
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
]
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {

```

```
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    }
}
```

```
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    },
    "timeoutNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        }
    }
},
```

```
"intent": {
    "name": "string",
    "slots": {
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
      ],
    }
  },
  "confirmationConditional": {
    "active": boolean,
    "conditionalBranches": [
      {
        "condition": {
          "expressionString": "string"
        },
        "name": "string",
        "nextStep": {
          "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
          },
          "intent": {
            "name": "string",
            "slots": {
              "string" : {
                "shape": "string",
                "value": {
                  "interpretedValue": "string"
                },
                "values": [
                  "SlotValueOverride"
                ]
              }
            }
          }
        },
        "sessionAttributes": {
          "string" : "string"
        }
      },
      "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
          {
            "message": {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              },
              "plainTextMessage": {
                "value": "string"
              },
              "ssmlMessage": {
                "value": "string"
              }
            },
            "variations": [

```

```
        {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        }
      }
    ]
  }
}
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  }
},
"response": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ]
        }
      }
    }
  ]
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
],
"confirmationNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
```

```
"confirmationResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "declinationConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                        "type": "string"
                    }
                }
            }
        ]
    }
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    }
}
```

```
        }
      ]
    }
  ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [

```

```
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
},
"declinationNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"declinationResponse": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        }
      },
      "plainTextMessage": {

```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
{
        "condition": {
            "expressionString": "string"
        },
        "name": "string",
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {
                "name": "string",
                "slots": {
                    "string" : {
                        "shape": "string",
                        "value": {
                            "interpretedValue": "string"
                        },
                        "values": [
                            "SlotValueOverride"
                        ]
                    }
                }
            }
        }
    }
},
"sessionAttributes": {
```

```
        "string" : "string"
    },
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
```

```
"name": "string",
"slots": {
    "string" : {
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ]
      }
    ]
  }
},
"failureNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"failureResponse": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      },
      "variations": [
        {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"promptSpecification": {
    "allowInterrupt": boolean,
    "maxRetries": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
      ],
    ],
  },
  "messageSelectionStrategy": "string",
  "promptAttemptsSpecification": {
    "string" : {
      "allowedInputTypes": {
        "allowAudioInput": boolean,
        "allowDTMFInput": boolean
      },
      "allowInterrupt": boolean,
      "audioAndDTMFInputSpecification": {
        "audioSpecification": {
          "endTimeTimeoutMs": number,
          "maxLengthMs": number
        },
        "dtmfSpecification": {
          "deletionCharacter": "string",
          "endCharacter": "string",
          "endTimeTimeoutMs": number,
          "maxLength": number
        },
        "startTimeTimeoutMs": number
      },
      "textInputSpecification": {
        "startTimeTimeoutMs": number
      }
    }
  }
},
"intentId": "string",
"intentName": "string",
"kendraConfiguration": {
  "kendraIndex": "string",
  "queryFilterString": "string",
  "queryFilterStringEnabled": boolean
},
"localeId": "string",
"outputContexts": [
  {
    "name": "string",
    "timeToLiveInSeconds": number,
    "turnsToLive": number
  }
],
"parentIntentSignature": "string",
"sampleUtterances": [
  {
    "utterance": "string"
  }
]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId (p. 104)**

The identifier of the bot associated with the intent.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [botVersion \(p. 104\)](#)

The identifier of the version of the bot associated with the intent.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

#### [creationDateTime \(p. 104\)](#)

A timestamp of the date and time that the intent was created.

Type: Timestamp

#### [description \(p. 104\)](#)

The description specified for the intent.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

#### [dialogCodeHook \(p. 104\)](#)

The dialog Lambda function specified for the intent.

Type: [DialogCodeHookSettings \(p. 792\)](#) object

#### [fulfillmentCodeHook \(p. 104\)](#)

The fulfillment Lambda function specified for the intent.

Type: [FulfillmentCodeHookSettings \(p. 804\)](#) object

#### [initialResponseSetting \(p. 104\)](#)

Configuration settings for the response that is sent to the user at the beginning of a conversation, before eliciting slot values.

Type: [InitialResponseSetting \(p. 817\)](#) object

#### [inputContexts \(p. 104\)](#)

The list of input contexts specified for the intent.

Type: Array of [InputContext \(p. 818\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

#### [intentClosingSetting \(p. 104\)](#)

The closing setting specified for the intent.

Type: [IntentClosingSetting \(p. 819\)](#) object

#### [intentConfirmationSetting \(p. 104\)](#)

The confirmation setting specified for the intent.

Type: [IntentConfirmationSetting \(p. 820\)](#) object

### [intentId \(p. 104\)](#)

A unique identifier for the intent.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [intentName \(p. 104\)](#)

The name specified for the intent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

### [kendraConfiguration \(p. 104\)](#)

Configuration for searching a Amazon Kendra index specified for the intent.

Type: [KendraConfiguration \(p. 829\)](#) object

### [localeId \(p. 104\)](#)

The locale that the intent is specified to use.

Type: String

### [outputContexts \(p. 104\)](#)

The list of output contexts specified for the intent.

Type: Array of [OutputContext \(p. 837\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

### [parentIntentSignature \(p. 104\)](#)

The signature of the parent intent specified for the intent.

Type: String

### [sampleUtterances \(p. 104\)](#)

The sample utterances specified for the intent.

Type: Array of [SampleUtterance \(p. 854\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateResourcePolicy

Service: Amazon Lex Model Building V2

Creates a new resource policy with the specified policy statements.

### Request Syntax

```
POST /policy/resourceArn HTTP/1.1
Content-type: application/json

{
    "policy": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **resourceArn** (p. 163)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **policy** (p. 163)

A resource policy to add to the resource. The policy is a JSON structure that contains one or more statements that define the policy. The policy must follow the IAM syntax. For more information about the contents of a JSON policy document, see [IAM JSON policy reference](#).

If the policy isn't valid, Amazon Lex returns a validation exception.

Type: String

Length Constraints: Minimum length of 2.

Required: Yes

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "resourceArn": "string",
    "revisionId": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### **resourceArn (p. 163)**

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy was attached to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

#### **revisionId (p. 163)**

The current revision of the resource policy. Use the revision ID to make sure that you are updating the most current version of a resource policy when you add a policy statement to a resource, delete a resource, or update a resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateResourcePolicyStatement

Service: Amazon Lex Model Building V2

Adds a new resource policy statement to a bot or bot alias. If a resource policy exists, the statement is added to the current resource policy. If a policy doesn't exist, a new policy is created.

You can't create a resource policy statement that allows cross-account access.

### Request Syntax

```
POST /policy/resourceArn/statements/?expectedRevisionId=expectedRevisionId HTTP/1.1
Content-type: application/json

{
  "action": [ "string" ],
  "condition": {
    "string": {
      "string": "string"
    }
  },
  "effect": "string",
  "principal": [
    {
      "arn": "string",
      "service": "string"
    }
  ],
  "statementId": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### [expectedRevisionId \(p. 166\)](#)

The identifier of the revision of the policy to edit. If this revision ID doesn't match the current revision ID, Amazon Lex throws an exception.

If you don't specify a revision, Amazon Lex overwrites the contents of the policy with the new values.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

#### [resourceArn \(p. 166\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### [action \(p. 166\)](#)

The Amazon Lex action that this policy either allows or denies. The action must apply to the resource type of the specified ARN. For more information, see [Actions, resources, and condition keys for Amazon Lex V2](#).

Type: Array of strings

Length Constraints: Minimum length of 5. Maximum length of 50.

Pattern: lex:[a-zA-Z\*]+\$

Required: Yes

#### [condition \(p. 166\)](#)

Specifies a condition when the policy is in effect. If the principal of the policy is a service principal, you must provide two condition blocks, one with a SourceAccount global condition key and one with a SourceArn global condition key.

For more information, see [IAM JSON policy elements: Condition](#).

Type: String to string to string map map

Map Entries: Minimum number of 0 items. Maximum number of 10 items.

Key Length Constraints: Minimum length of 1.

Map Entries: Minimum number of 0 items. Maximum number of 10 items.

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

Value Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

#### [effect \(p. 166\)](#)

Determines whether the statement allows or denies access to the resource.

Type: String

Valid Values: Allow | Deny

Required: Yes

#### [principal \(p. 166\)](#)

An IAM principal, such as an IAM users, IAM roles, or AWS services that is allowed or denied access to a resource. For more information, see [AWS JSON policy elements: Principal](#).

Type: Array of [Principal \(p. 844\)](#) objects

Required: Yes

#### [statementId \(p. 166\)](#)

The name of the statement. The ID is the same as the Sid IAM property. The statement name must be unique within the policy. For more information, see [IAM JSON policy elements: Sid](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

## [Response Syntax](#)

HTTP/1.1 200

```
Content-type: application/json

{
    "resourceArn": "string",
    "revisionId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [resourceArn \(p. 167\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

### [revisionId \(p. 167\)](#)

The current revision of the resource policy. Use the revision ID to make sure that you are updating the most current version of a resource policy when you add a policy statement to a resource, delete a resource, or update a resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateSlot

Service: Amazon Lex Model Building V2

Creates a slot in an intent. A slot is a variable needed to fulfill an intent. For example, an OrderPizza intent might need slots for size, crust, and number of pizzas. For each slot, you define one or more utterances that Amazon Lex uses to elicit a response from the user.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/slots/ HTTP/1.1
Content-type: application/json
```

```
{
    "description": "string",
    "multipleValuesSetting": {
        "allowMultipleValues": boolean
    },
    "obfuscationSetting": {
        "obfuscationSettingType": "string"
    },
    "slotName": "string",
    "slotTypeId": "string",
    "subSlotSetting": {
        "expression": "string",
        "slotSpecifications": {
            "string": {
                "slotTypeId": "string",
                "valueElicitationSetting": {
                    "defaultValueSpecification": {
                        "defaultValueList": [
                            {
                                "defaultValue": "string"
                            }
                        ]
                    },
                    "promptSpecification": {
                        "allowInterrupt": boolean,
                        "maxRetries": number,
                        "messageGroups": [
                            {
                                "message": {
                                    "customPayload": {
                                        "value": "string"
                                    },
                                    "imageResponseCard": {
                                        "buttons": [
                                            {
                                                "text": "string",
                                                "value": "string"
                                            }
                                        ],
                                        "imageUrl": "string",
                                        "subtitle": "string",
                                        "title": "string"
                                    },
                                    "plainTextMessage": {
                                        "value": "string"
                                    },
                                    "ssmlMessage": {
                                        "value": "string"
                                    }
                                },
                                "variations": [
                                    {

```

```
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
],
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeTimeoutMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeTimeoutMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "text": "string"
                }
            }
        ]
    }
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {

```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
],
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
{
    "message": {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
{
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
}
},
"valueElicitationSetting": {
    "defaultValueSpecification": {
        "defaultValueList": [
            {
                "defaultValue": "string"
            }
        ]
    },
    "promptSpecification": {
        "allowInterrupt": boolean,
        "maxRetries": number,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },

```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
],
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"slotCaptureSetting": {
    "captureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                    }
                }
            }
        ]
    }
}
```

```
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {

```

```
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {

```

```
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    }
  ]
}
},
"captureNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"captureResponse": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        }
      }
    }
  ]
}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"codeHook": {
    "active": boolean,
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string",
    "postCodeHookSpecification": {
        "failureConditional": {
            "active": boolean,
            "conditionalBranches": [
                {
                    "condition": {
                        "expressionString": "string"
                    },
                    "name": "string",
                    "nextStep": {
                        "dialogAction": {
                            "slotToElicit": "string",
                            "suppressNextMessage": boolean,
                            "type": "string"
                        },
                        "intent": {
                            "name": "string",
                            "slots": {
                                "string" : {
                                    "shape": "string",
                                    "value": {
                                        "interpretedValue": "string"
                                    }
                                },
                                "values": [
                                    "SlotValueOverride"
                                ]
                            }
                        }
                    }
                }
            ]
        }
    }
}
```

```
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ]
    },
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            }
        }
    }
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    }
}
```

```
        }
      ]
    }
  ],
  "failureNextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
}
```

```
        "text": "string",
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            }
        }
    ],
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "image": "string"
                }
            }
        }
    ]
}
```

```
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
```

```
"allowInterrupt": boolean,
"messageGroups": [
  {
    "message": {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  }
],
"successNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string": {
        "shape": "string",
        "value": {
          "string": "string"
        }
      }
    }
  }
}
```

```
        "interpretedValue": "string"
    },
    "values": [
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "timeoutConditional": {
```

```
"active": boolean,
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
]
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
{
    "timeoutNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string": {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string": "string"
        }
    },
    "timeoutResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            }
        ]
    }
}
```

```
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {

```

```
"name": "string",
"slots": {
    "string" : {
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ]
      }
    ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": [
              "interpretedValue": "string"
            ],
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              }
            }
          ]
        }
      ]
    }
  }
}
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
            }
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
{
    "slotConstraint": "string",
    "waitAndContinueSpecification": {
        "active": boolean,
        "continueResponse": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            ]
        }
    }
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {

```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
],
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
}
```

}

## URI Request Parameters

The request uses the following URI parameters.

### **botId (p. 170)**

The identifier of the bot associated with the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **botVersion (p. 170)**

The version of the bot associated with the slot.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

### **intentId (p. 170)**

The identifier of the intent that contains the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **localeId (p. 170)**

The identifier of the language and locale that the slot will be used in. The string must match one of the supported locales. All of the bots, intents, slot types used by the slot must have the same locale. For more information, see [Supported languages](#).

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### **description (p. 170)**

A description of the slot. Use this to help identify the slot in lists.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

### **multipleValuesSetting (p. 170)**

Indicates whether the slot returns multiple values in one response. Multi-value slots are only available in the en-US locale. If you set this value to true in any other locale, Amazon Lex throws a ValidationException.

If the `multipleValuesSetting` is not set, the default value is `false`.

Type: [MultipleValuesSetting \(p. 834\)](#) object

Required: No

#### [obfuscationSetting \(p. 170\)](#)

Determines how slot values are used in Amazon CloudWatch logs. If the value of the `obfuscationSetting` parameter is `DefaultObfuscation`, slot values are obfuscated in the log output. If the value is `None`, the actual value is present in the log output.

The default is to obfuscate values in the CloudWatch logs.

Type: [ObfuscationSetting \(p. 836\)](#) object

Required: No

#### [slotName \(p. 170\)](#)

The name of the slot. Slot names must be unique within the bot that contains the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^([0-9a-zA-Z][_-]?)+$`

Required: Yes

#### [slotTypeId \(p. 170\)](#)

The unique identifier for the slot type associated with this slot. The slot type determines the values that can be entered into the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern: `^(AMAZON\.)[a-zA-Z_]+?|[0-9a-zA-Z]+)$`

Required: No

#### [subSlotSetting \(p. 170\)](#)

Specifications for the constituent sub slots and the expression for the composite slot.

Type: [SubSlotSetting \(p. 881\)](#) object

Required: No

#### [valueElicitationSetting \(p. 170\)](#)

Specifies prompts that Amazon Lex sends to the user to elicit a response that provides the value for the slot.

Type: [SlotValueElicitationSetting \(p. 873\)](#) object

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{  
    "botId": "string",  
    "botVersion": "string",  
    "creationDateTime": number,  
    "description": "string",  
    "intentId": "string",  
    "localeId": "string",  
    "multipleValuesSetting": {  
        "allowMultipleValues": boolean  
    },  
    "obfuscationSetting": {  
        "obfuscationSettingType": "string"  
    },  
    "slotId": "string",  
    "slotName": "string",  
    "slotTypeId": "string",  
    "subSlotSetting": {  
        "expression": "string",  
        "slotSpecifications": {  
            "string" : {  
                "slotTypeId": "string",  
                "valueElicitationSetting": {  
                    "defaultValueSpecification": {  
                        "defaultValueList": [  
                            {  
                                "defaultValue": "string"  
                            }  
                        ]  
                    },  
                    "promptSpecification": {  
                        "allowInterrupt": boolean,  
                        "maxRetries": number,  
                        "messageGroups": [  
                            {  
                                "message": {  
                                    "customPayload": {  
                                        "value": "string"  
                                    },  
                                    "imageResponseCard": {  
                                        "buttons": [  
                                            {  
                                                "text": "string",  
                                                "value": "string"  
                                            }  
                                        ],  
                                        "imageUrl": "string",  
                                        "subtitle": "string",  
                                        "title": "string"  
                                    },  
                                    "plainTextMessage": {  
                                        "value": "string"  
                                    },  
                                    "ssmlMessage": {  
                                        "value": "string"  
                                    }  
                                },  
                                "variations": [  
                                    {  
                                        "customPayload": {  
                                            "value": "string"  
                                        },  
                                        "imageResponseCard": {  
                                            "buttons": [  
                                                {  
                                                    "text": "string",  
                                                    "value": "string"  
                                                }  
                                            ]  
                                        }  
                                    }  
                                ]  
                            }  
                        ]  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeoutMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeoutMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
{
    "utterance": "string"
}
],
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "maxWidth": number,
                        "maxHeight": number
                    }
                }
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
{
    "message": {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
{

```

```
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
}
},
"valueElicitationSetting": {
    "defaultValueSpecification": {
        "defaultValueList": [
            {
                "defaultValue": "string"
            }
        ]
    },
    "promptSpecification": {
        "allowInterrupt": boolean,
        "maxRetries": number,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeTimeoutMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeTimeoutMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"slotCaptureSetting": {
    "captureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                        "type": "string"
                    },
                    "intent": {
                        "name": "string",
                        "slots": {
                            "string" : {
                                "shape": "string",
                                "value": {

```

```
        "interpretedValue": "string"
    },
    "values": [
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"captureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"captureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [

```

```
{  
    "customPayload": {  
        "value": "string"  
    },  
    "imageResponseCard": {  
        "buttons": [  
            {  
                "text": "string",  
                "value": "string"  
            }  
        ],  
        "imageUrl": "string",  
        "subtitle": "string",  
        "title": "string"  
    },  
    "plainTextMessage": {  
        "value": "string"  
    },  
    "ssmlMessage": {  
        "value": "string"  
    }  
},  
]  
}  
]  
},  
"codeHook": {  
    "active": boolean,  
    "enableCodeHookInvocation": boolean,  
    "invocationLabel": "string",  
    "postCodeHookSpecification": {  
        "failureConditional": {  
            "active": boolean,  
            "conditionalBranches": [  
                {  
                    "condition": {  
                        "expressionString": "string"  
                    },  
                    "name": "string",  
                    "nextStep": {  
                        "dialogAction": {  
                            "slotToElicit": "string",  
                            "suppressNextMessage": boolean,  
                            "type": "string"  
                        },  
                        "intent": {  
                            "name": "string",  
                            "slots": {  
                                "string" : {  
                                    "shape": "string",  
                                    "value": {  
                                        "interpretedValue": "string"  
                                    },  
                                    "values": [  
                                        "SlotValueOverride"  
                                    ]  
                                }  
                            }  
                        },  
                        "sessionAttributes": {  
                            "string" : "string"  
                        }  
                    },  
                    "response": {  
                        "allowInterrupt": boolean,  
                        "messageGroups": [  
                            {  
                                "content": "string",  
                                "type": "string"  
                            }  
                        ]  
                    }  
                }  
            ]  
        }  
    }  
}
```

```
{  
    "message": {  
        "customPayload": {  
            "value": "string"  
        },  
        "imageResponseCard": {  
            "buttons": [  
                {  
                    "text": "string",  
                    "value": "string"  
                }  
            ],  
            "imageUrl": "string",  
            "subtitle": "string",  
            "title": "string"  
        },  
        "plainTextMessage": {  
            "value": "string"  
        },  
        "ssmlMessage": {  
            "value": "string"  
        }  
    },  
    "variations": [  
        {  
            "customPayload": {  
                "value": "string"  
            },  
            "imageResponseCard": {  
                "buttons": [  
                    {  
                        "text": "string",  
                        "value": "string"  
                    }  
                ],  
                "imageUrl": "string",  
                "subtitle": "string",  
                "title": "string"  
            },  
            "plainTextMessage": {  
                "value": "string"  
            },  
            "ssmlMessage": {  
                "value": "string"  
            }  
        }  
    ]  
},  
"defaultBranch": {  
    "nextStep": {  
        "dialogAction": {  
            "slotToElicit": "string",  
            "suppressNextMessage": boolean,  
            "type": "string"  
        },  
        "intent": {  
            "name": "string",  
            "slots": {  
                "string" : {  
                    "shape": "string",  
                    "value": {  
                        "interpretedValue": "string"  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
```

```
"failureNextStep": {  
    "dialogAction": {  
        "slotToElicit": "string",  
        "suppressNextMessage": boolean,  
        "type": "string"  
    },  
    "intent": {  
        "name": "string",  
        "slots": {  
            "string" : {  
                "shape": "string",  
                "value": {  
                    "interpretedValue": "string"  
                },  
                "values": [  
                    "SlotValueOverride"  
                ]  
            }  
        }  
    },  
    "sessionAttributes": {  
        "string" : "string"  
    }  
},  
"failureResponse": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "text": "string"  
                }  
            ]  
        }  
    ]  
},  
"idleSessionTTL": integer,  
"idleSessionType": "string",  
"maxConcurrentSessions": integer,  
"maxInputLength": integer,  
"maxOutputLength": integer,  
"maxResponseCardSize": integer,  
"maxTextMessageLength": integer,  
"maxVariations": integer,  
"name": "string",  
"idleSessionMinutes": integer,  
"idleSessionType": "string",  
"idleSessionTTL": integer,  
"maxConcurrentSessions": integer,  
"maxInputLength": integer,  
"maxOutputLength": integer,  
"maxResponseCardSize": integer,  
"maxTextMessageLength": integer,  
"maxVariations": integer,  
"name": "string",  
"idleSessionMinutes": integer}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {

```

```
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string": {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
},
```

```
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "timeoutConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {

```

```
"dialogAction": {  
    "slotToElicit": "string",  
    "suppressNextMessage": boolean,  
    "type": "string"  
},  
"intent": {  
    "name": "string",  
    "slots": {  
        "string" : {  
            "shape": "string",  
            "value": {  
                "interpretedValue": "string"  
            },  
            "values": [  
                "SlotValueOverride"  
            ]  
        }  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"response": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            }
                        },
                        "values": [

```

```
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "defaultBranch": {
        "nextStep": {

```

```
"dialogAction": {  
    "slotToElicit": "string",  
    "suppressNextMessage": boolean,  
    "type": "string"  
},  
"intent": {  
    "name": "string",  
    "slots": {  
        "string" : {  
            "shape": "string",  
            "value": {  
                "interpretedValue": "string"  
            },  
            "values": [  
                "SlotValueOverride"  
            ]  
        }  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"response": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
}
},
"slotConstraint": "string",
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ],
        "timeoutInSeconds": number
    },
    "waitingResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**botId (p. 198)**

The unique identifier of the bot associated with the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botVersion (p. 198)**

The version of the bot associated with the slot.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

**creationDateTime (p. 198)**

The timestamp of the date and time that the slot was created.

Type: Timestamp

**description (p. 198)**

The description associated with the slot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**intentId (p. 198)**

The unique identifier of the intent associated with the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**localeId (p. 198)**

The language and local specified for the slot.

Type: String

**multipleValuesSetting (p. 198)**

Indicates whether the slot returns multiple values in one response.

Type: [MultipleValuesSetting \(p. 834\)](#) object

**obfuscationSetting (p. 198)**

Indicates whether the slot is configured to obfuscate values in Amazon CloudWatch logs.

Type: [ObfuscationSetting \(p. 836\)](#) object

**slotId (p. 198)**

The unique identifier associated with the slot. Use this to identify the slot when you update or delete it.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**slotName (p. 198)**

The name specified for the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

**slotTypeId (p. 198)**

The unique identifier of the slot type associated with this slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern: ^((AMAZON\.)[a-zA-Z\_]+?|[0-9a-zA-Z]+)\$

**subSlotSetting (p. 198)**

Specifications for the constituent sub slots and the expression for the composite slot.

Type: [SubSlotSetting \(p. 881\)](#) object

**valueElicitationSetting (p. 198)**

The value elicitation settings specified for the slot.

Type: [SlotValueElicitationSetting \(p. 873\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

**ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

**InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateSlotType

Service: Amazon Lex Model Building V2

Creates a custom slot type

To create a custom slot type, specify a name for the slot type and a set of enumeration values, the values that a slot of this type can assume.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/slottypes/ HTTP/1.1
Content-type: application/json

{
  "compositeSlotTypeSetting": {
    "subSlots": [
      {
        "name": "string",
        "slotTypeId": "string"
      }
    ]
  },
  "description": "string",
  "externalSourceSetting": {
    "grammarSlotTypeSetting": {
      "source": {
        "kmsKeyArn": "string",
        "s3BucketName": "string",
        "s3ObjectKey": "string"
      }
    }
  },
  "parentSlotTypeSignature": "string",
  "slotTypeName": "string",
  "slotTypeValues": [
    {
      "sampleValue": {
        "value": "string"
      },
      "synonyms": [
        {
          "value": "string"
        }
      ]
    }
  ],
  "valueSelectionSetting": {
    "advancedRecognitionSetting": {
      "audioRecognitionStrategy": "string"
    },
    "regexFilter": {
      "pattern": "string"
    },
    "resolutionStrategy": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

### **botId (p. 229)**

The identifier of the bot associated with this slot type.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **botVersion (p. 229)**

The identifier of the bot version associated with this slot type.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

### **localeId (p. 229)**

The identifier of the language and locale that the slot type will be used in. The string must match one of the supported locales. All of the bots, intents, and slots used by the slot type must have the same locale. For more information, see [Supported languages](#).

Required: Yes

## **Request Body**

The request accepts the following data in JSON format.

### **compositeSlotTypeSetting (p. 229)**

Specifications for a composite slot type.

Type: [CompositeSlotTypeSetting \(p. 777\)](#) object

Required: No

### **description (p. 229)**

A description of the slot type. Use the description to help identify the slot type in lists.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

### **externalSourceSetting (p. 229)**

Sets the type of external information used to create the slot type.

Type: [ExternalSourceSetting \(p. 802\)](#) object

Required: No

### **parentSlotTypeSignature (p. 229)**

The built-in slot type used as a parent of this slot type. When you define a parent slot type, the new slot type has the configuration of the parent slot type.

Only AMAZON.AlphaNumeric is supported.

Type: String

Required: No

#### [slotTypeName \(p. 229\)](#)

The name for the slot. A slot type name must be unique within the account.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### [slotTypeValues \(p. 229\)](#)

A list of SlotTypeValue objects that defines the values that the slot type can take. Each value can have a list of synonyms, additional values that help train the machine learning model about the values that it resolves for a slot.

Type: Array of [SlotTypeValue \(p. 871\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10000 items.

Required: No

#### [valueSelectionSetting \(p. 229\)](#)

Determines the strategy that Amazon Lex uses to select a value from the list of possible values. The field can be set to one of the following values:

- **OriginalValue** - Returns the value entered by the user, if the user value is similar to the slot value.
- **TopResolution** - If there is a resolution list for the slot, return the first value in the resolution list. If there is no resolution list, return null.

If you don't specify the valueSelectionSetting parameter, the default is OriginalValue.

Type: [SlotValueSelectionSetting \(p. 877\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "compositeSlotTypeSetting": {
        "subSlots": [
            {
                "name": "string",
                "slotTypeId": "string"
            }
        ]
    },
    "creationDateTime": number,
    "description": "string",
    "externalSourceSetting": {
```

```
        "grammarSlotTypeSetting": {
            "source": {
                "kmsKeyArn": "string",
                "s3BucketName": "string",
                "s3ObjectKey": "string"
            }
        },
        "localeId": "string",
        "parentSlotTypeSignature": "string",
        "slotTypeId": "string",
        "slotTypeName": "string",
        "slotTypeValues": [
            {
                "sampleValue": {
                    "value": "string"
                },
                "synonyms": [
                    {
                        "value": "string"
                    }
                ]
            }
        ],
        "valueSelectionSetting": {
            "advancedRecognitionSetting": {
                "audioRecognitionStrategy": "string"
            },
            "regexFilter": {
                "pattern": "string"
            },
            "resolutionStrategy": "string"
        }
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId** (p. 231)

The identifier for the bot associated with the slot type.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion** (p. 231)

The version of the bot associated with the slot type.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **compositeSlotTypeSetting** (p. 231)

Specifications for a composite slot type.

Type: [CompositeSlotTypeSetting \(p. 777\)](#) object

**creationDateTime (p. 231)**

A timestamp of the date and time that the slot type was created.

Type: Timestamp

**description (p. 231)**

The description specified for the slot type.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**externalSourceSetting (p. 231)**

The type of external information used to create the slot type.

Type: [ExternalSourceSetting \(p. 802\)](#) object

**localeId (p. 231)**

The specified language and local specified for the slot type.

Type: String

**parentSlotTypeSignature (p. 231)**

The signature of the base slot type specified for the slot type.

Type: String

**slotTypeId (p. 231)**

The unique identifier assigned to the slot type. Use this to identify the slot type in the `UpdateSlotType` and `DeleteSlotType` operations.

Type: String

Length Constraints: Fixed length of 10.

Pattern: `^[0-9a-zA-Z]+$`

**slotTypeName (p. 231)**

The name specified for the slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^([0-9a-zA-Z][_-]?)+$`

**slotTypeValues (p. 231)**

The list of values that the slot type can assume.

Type: Array of [SlotTypeValue \(p. 871\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10000 items.

**valueSelectionSetting (p. 231)**

The strategy that Amazon Lex uses to select a value from the list of possible values.

Type: [SlotValueSelectionSetting \(p. 877\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateUploadUrl

Service: Amazon Lex Model Building V2

Gets a pre-signed S3 write URL that you use to upload the zip archive when importing a bot or a bot locale.

### Request Syntax

```
POST /createuploadurl/ HTTP/1.1
```

### URI Request Parameters

The request does not use any URI parameters.

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "importId": "string",
    "uploadUrl": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### [importId \(p. 235\)](#)

An identifier for a unique import job. Use it when you call the [StartImport](#) operation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [uploadUrl \(p. 235\)](#)

A pre-signed S3 write URL. Upload the zip archive file that contains the definition of your bot or bot locale.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

### Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteBot

Service: Amazon Lex Model Building V2

Deletes all versions of a bot, including the Draft version. To delete a specific version, use the DeleteBotVersion operation.

When you delete a bot, all of the resources contained in the bot are also deleted. Deleting a bot removes all locales, intents, slot, and slot types defined for the bot.

If a bot has an alias, the DeleteBot operation returns a ResourceInUseException exception. If you want to delete the bot and the alias, set the skipResourceInUseCheck parameter to true.

### Request Syntax

```
DELETE /bots/botId?skipResourceInUseCheck=skipResourceInUseCheck HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 237)

The identifier of the bot to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **skipResourceInUseCheck** (p. 237)

When true, Amazon Lex doesn't check to see if another resource, such as an alias, is using the bot before it is deleted.

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "botId": "string",
  "botStatus": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

#### **botId** (p. 237)

The unique identifier of the bot that Amazon Lex is deleting.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [botStatus \(p. 237\)](#)

The current status of the bot. The status is Deleting while the bot and its associated resources are being deleted.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteBotAlias

Service: Amazon Lex Model Building V2

Deletes the specified bot alias.

### Request Syntax

```
DELETE /bots/botId/botaliases/botAliasId?skipResourceInUseCheck=skipResourceInUseCheck
HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botAliasId** (p. 240)

The unique identifier of the bot alias to delete.

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

Required: Yes

#### **botId** (p. 240)

The unique identifier of the bot associated with the alias to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **skipResourceInUseCheck** (p. 240)

When this parameter is true, Amazon Lex doesn't check to see if any other resource is using the alias before it is deleted.

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "botAliasId": "string",
  "botAliasStatus": "string",
  "botId": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### [botAliasId \(p. 240\)](#)

The unique identifier of the bot alias to delete.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

### [botAliasStatus \(p. 240\)](#)

The current status of the alias. The status is Deleting while the alias is in the process of being deleted. Once the alias is deleted, it will no longer appear in the list of aliases returned by the ListBotAliases operation.

Type: String

Valid Values: Creating | Available | Deleting | Failed

### [botId \(p. 240\)](#)

The unique identifier of the bot that contains the alias to delete.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### ValidationException

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteBotLocale

Service: Amazon Lex Model Building V2

Removes a locale from a bot.

When you delete a locale, all intents, slots, and slot types defined for the locale are also deleted.

### Request Syntax

```
DELETE /bots/botId/botversions/botVersion/botlocales/localeId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 243)

The unique identifier of the bot that contains the locale.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 243)

The version of the bot that contains the locale.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 243)

The identifier of the language and locale that will be deleted. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botId": "string",
    "botLocaleStatus": "string",
    "botVersion": "string",
    "localeId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### **botId (p. 243)**

The identifier of the bot that contained the deleted locale.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botLocaleStatus (p. 243)**

The status of deleting the bot locale. The locale first enters the Deleting status. Once the locale is deleted it no longer appears in the list of locales for the bot.

Type: String

Valid Values: Creating | Building | Built | ReadyExpressTesting | Failed | Deleting | NotBuilt | Importing | Processing

### **botVersion (p. 243)**

The version of the bot that contained the deleted locale.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **localeId (p. 243)**

The language and locale of the deleted locale.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteBotVersion

Service: Amazon Lex Model Building V2

Deletes a specific version of a bot. To delete all version of a bot, use the [DeleteBot](#) operation.

### Request Syntax

```
DELETE /bots/botId/botversions/botVersion?skipResourceInUseCheck=skipResourceInUseCheck
HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### [botId](#) (p. 246)

The identifier of the bot that contains the version.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### [botVersion](#) (p. 246)

The version of the bot to delete.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

Required: Yes

#### [skipResourceInUseCheck](#) (p. 246)

By default, the DeleteBotVersion operations throws a ResourceInUseException exception if you try to delete a bot version that has an alias pointing at it. Set the skipResourceInUseCheck parameter to true to skip this check and remove the version even if an alias points to it.

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "botId": "string",
  "botStatus": "string",
  "botVersion": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

**botId (p. 246)**

The identifier of the bot that is being deleted.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botStatus (p. 246)**

The current status of the bot.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

**botVersion (p. 246)**

The version of the bot that is being deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

**ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

**InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteCustomVocabulary

Service: Amazon Lex Model Building V2

Removes a custom vocabulary from the specified locale in the specified bot.

### Request Syntax

```
DELETE /bots/botId/botversions/botVersion/botlocales/localeId/customvocabulary HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 249)

The unique identifier of the bot to remove the custom vocabulary from.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 249)

The version of the bot to remove the custom vocabulary from.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 249)

The locale identifier for the locale that contains the custom vocabulary to remove.

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "customVocabularyStatus": "string",
    "localeId": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

**botId (p. 249)**

The identifier of the bot that the custom vocabulary was removed from.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botVersion (p. 249)**

The version of the bot that the custom vocabulary was removed from.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

**customVocabularyStatus (p. 249)**

The status of removing the custom vocabulary.

Type: String

Valid Values: Ready | Deleting | Exporting | Importing | Creating

**localeId (p. 249)**

The locale identifier for the locale that the custom vocabulary was removed from.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

**ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

**InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteExport

Service: Amazon Lex Model Building V2

Removes a previous export and the associated files stored in an S3 bucket.

### Request Syntax

```
DELETE /exports/exportId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **exportId** (p. 252)

The unique identifier of the export to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "exportId": "string",
    "exportStatus": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

#### **exportId** (p. 252)

The unique identifier of the deleted export.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### **exportStatus** (p. 252)

The current status of the deletion. When the deletion is complete, the export will no longer be returned by the [ListExports](#) operation and calls to the [DescribeExport](#) operation with the export identifier will fail.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteImport

Service: Amazon Lex Model Building V2

Removes a previous import and the associated file stored in an S3 bucket.

### Request Syntax

```
DELETE /imports/importId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **importId (p. 254)**

The unique identifier of the import to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "importId": "string",
    "importStatus": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

#### **importId (p. 254)**

The unique identifier of the deleted import.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### **importStatus (p. 254)**

The current status of the deletion. When the deletion is complete, the import will no longer be returned by the [ListImports](#) operation and calls to the [DescribeImport](#) operation with the import identifier will fail.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteIntent

Service: Amazon Lex Model Building V2

Removes the specified intent.

Deleting an intent also deletes the slots associated with the intent.

### Request Syntax

```
DELETE /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 256)

The identifier of the bot associated with the intent.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 256)

The version of the bot associated with the intent.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **intentId** (p. 256)

The unique identifier of the intent to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **localeId** (p. 256)

The identifier of the language and locale where the bot will be deleted. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 204
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## DeleteResourcePolicy

Service: Amazon Lex Model Building V2

Removes an existing policy from a bot or bot alias. If the resource doesn't have a policy attached, Amazon Lex returns an exception.

### Request Syntax

```
DELETE /policy/resourceArn?expectedRevisionId=expectedRevisionId HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### [expectedRevisionId \(p. 259\)](#)

The identifier of the revision to edit. If this ID doesn't match the current revision number, Amazon Lex returns an exception

If you don't specify a revision ID, Amazon Lex will delete the current policy.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

#### [resourceArn \(p. 259\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that has the resource policy attached.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 204
Content-type: application/json

{
  "resourceArn": "string",
  "revisionId": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 204 response.

The following data is returned in JSON format by the service.

#### [resourceArn \(p. 259\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy was deleted from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

#### **revisionId (p. 259)**

The current revision of the resource policy. Use the revision ID to make sure that you are updating the most current version of a resource policy when you add a policy statement to a resource, delete a resource, or update a resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteResourcePolicyStatement

Service: Amazon Lex Model Building V2

Deletes a policy statement from a resource policy. If you delete the last statement from a policy, the policy is deleted. If you specify a statement ID that doesn't exist in the policy, or if the bot or bot alias doesn't have a policy attached, Amazon Lex returns an exception.

### Request Syntax

```
DELETE /policy/resourceArn/statements/statementId?expectedRevisionId=expectedRevisionId
HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### [expectedRevisionId \(p. 261\)](#)

The identifier of the revision of the policy to delete the statement from. If this revision ID doesn't match the current revision ID, Amazon Lex throws an exception.

If you don't specify a revision, Amazon Lex removes the current contents of the statement.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

#### [resourceArn \(p. 261\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

#### [statementId \(p. 261\)](#)

The name of the statement (SID) to delete from the policy.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 204
Content-type: application/json

{
    "resourceArn": "string",
    "revisionId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response.

The following data is returned in JSON format by the service.

### **resourceArn (p. 261)**

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy statement was removed from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

### **revisionId (p. 261)**

The current revision of the resource policy. Use the revision ID to make sure that you are updating the most current version of a resource policy when you add a policy statement to a resource, delete a resource, or update a resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteSlot

Service: Amazon Lex Model Building V2

Deletes the specified slot from an intent.

### Request Syntax

```
DELETE /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/  
slots/slotId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 264)

The identifier of the bot associated with the slot to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 264)

The version of the bot associated with the slot to delete.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **intentId** (p. 264)

The identifier of the intent associated with the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **localeId** (p. 264)

The identifier of the language and locale that the slot will be deleted from. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

#### **slotId** (p. 264)

The identifier of the slot to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 204
```

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteSlotType

Service: Amazon Lex Model Building V2

Deletes a slot type from a bot locale.

If a slot is using the slot type, Amazon Lex throws a `ResourceInUseException` exception. To avoid the exception, set the `skipResourceInUseCheck` parameter to true.

### Request Syntax

```
DELETE /bots/botId/botversions/botVersion/botlocales/localeId/slottypes/slotTypeId?  
skipResourceInUseCheck=skipResourceInUseCheck HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 267)

The identifier of the bot associated with the slot type.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 267)

The version of the bot associated with the slot type.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 267)

The identifier of the language and locale that the slot type will be deleted from. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

#### **skipResourceInUseCheck** (p. 267)

By default, the `DeleteSlotType` operations throws a `ResourceInUseException` exception if you try to delete a slot type used by a slot. Set the `skipResourceInUseCheck` parameter to true to skip this check and remove the slot type even if a slot uses it.

#### **slotTypeId** (p. 267)

The identifier of the slot type to delete.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

## Response Syntax

HTTP/1.1 204

## Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteUtterances

Service: Amazon Lex Model Building V2

Deletes stored utterances.

Amazon Lex stores the utterances that users send to your bot. Utterances are stored for 15 days for use with the [ListAggregatedUtterances](#) operation, and then stored indefinitely for use in improving the ability of your bot to respond to user input..

Use the DeleteUtterances operation to manually delete utterances for a specific session. When you use the DeleteUtterances operation, utterances stored for improving your bot's ability to respond to user input are deleted immediately. Utterances stored for use with the [ListAggregatedUtterances](#) operation are deleted after 15 days.

### Request Syntax

```
DELETE /bots/botId/utterances/?localeId=localeId&sessionId=sessionId HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 270)

The unique identifier of the bot that contains the utterances.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **localeId** (p. 270)

The identifier of the language and locale where the utterances were collected. The string must match one of the supported locales. For more information, see [Supported languages](#).

#### **sessionId** (p. 270)

The unique identifier of the session with the user. The ID is returned in the response from the [RecognizeText](#) and [RecognizeUtterance](#) operations.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 204
```

### Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeBot

Service: Amazon Lex Model Building V2

Provides metadata information about a bot.

### Request Syntax

```
GET /bots/botId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 272)

The unique identifier of the bot to describe.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botName": "string",
  "botStatus": "string",
  "creationDateTime": number,
  "dataPrivacy": {
    "childDirected": boolean
  },
  "description": "string",
  "idleSessionTTLInSeconds": number,
  "lastUpdatedDateTime": number,
  "roleArn": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### **botId** (p. 272)

The unique identifier of the bot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botName (p. 272)**

The name of the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

**botStatus (p. 272)**

The current status of the bot. When the status is Available the bot is ready to be used in conversations with users.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

**creationDateTime (p. 272)**

A timestamp of the date and time that the bot was created.

Type: Timestamp

**dataPrivacy (p. 272)**

Settings for managing data privacy of the bot and its conversations with users.

Type: [DataPrivacy \(p. 787\)](#) object

**description (p. 272)**

The description of the bot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**idleSessionTTLInSeconds (p. 272)**

The maximum time in seconds that Amazon Lex retains the data gathered in a conversation.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 86400.

**lastUpdatedDateTime (p. 272)**

A timestamp of the date and time that the bot was last updated.

Type: Timestamp

**roleArn (p. 272)**

The Amazon Resource Name (ARN) of an IAM role that has permission to access the bot.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:iam::[0-9]{12}:role/.+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeBotAlias

Service: Amazon Lex Model Building V2

Get information about a specific bot alias.

### Request Syntax

```
GET /bots/botId/botaliases/botAliasId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botAliasId** (p. 275)

The identifier of the bot alias to describe.

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

Required: Yes

#### **botId** (p. 275)

The identifier of the bot associated with the bot alias to describe.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botAliasHistoryEvents": [
    {
      "botVersion": "string",
      "endDate": number,
      "startDate": number
    }
  ],
  "botAliasId": "string",
  "botAliasLocaleSettings": {
    "string": {
      "codeHookSpecification": {
        "lambdaCodeHook": {
          "codeHookInterfaceVersion": "string",
          "lambdaARN": "string"
        }
      },
      "enabled": boolean
    }
  }
}
```

```
        },
        "botAliasName": "string",
        "botAliasStatus": "string",
        "botId": "string",
        "botVersion": "string",
        "conversationLogSettings": {
            "audioLogSettings": [
                {
                    "destination": {
                        "s3Bucket": {
                            "kmsKeyArn": "string",
                            "logPrefix": "string",
                            "s3BucketArn": "string"
                        }
                    },
                    "enabled": boolean
                }
            ],
            "textLogSettings": [
                {
                    "destination": {
                        "cloudWatch": {
                            "cloudWatchLogGroupArn": "string",
                            "logPrefix": "string"
                        }
                    },
                    "enabled": boolean
                }
            ]
        },
        "creationDateTime": number,
        "description": "string",
        "lastUpdatedDateTime": number,
        "sentimentAnalysisSettings": {
            "detectSentiment": boolean
        }
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botAliasHistoryEvents \(p. 275\)](#)

A list of events that affect a bot alias. For example, an event is recorded when the version that the alias points to changes.

Type: Array of [BotAliasHistoryEvent \(p. 744\)](#) objects

### [botAliasId \(p. 275\)](#)

The identifier of the bot alias.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

### [botAliasLocaleSettings \(p. 275\)](#)

The locale settings that are unique to the alias.

Type: String to [BotAliasLocaleSettings \(p. 745\)](#) object map

Map Entries: Maximum number of items.

**botAliasName (p. 275)**

The name of the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

**botAliasStatus (p. 275)**

The current status of the alias. When the alias is Available, the alias is ready for use with your bot.

Type: String

Valid Values: Creating | Available | Deleting | Failed

**botId (p. 275)**

The identifier of the bot associated with the bot alias.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botVersion (p. 275)**

The version of the bot associated with the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9])+\$

**conversationLogSettings (p. 275)**

Specifics of how Amazon Lex logs text and audio conversations with the bot associated with the alias.

Type: [ConversationLogSettings \(p. 781\)](#) object

**creationDateTime (p. 275)**

A timestamp of the date and time that the alias was created.

Type: Timestamp

**description (p. 275)**

The description of the bot alias.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**lastUpdatedDateTime (p. 275)**

A timestamp of the date and time that the alias was last updated.

Type: Timestamp

### [sentimentAnalysisSettings \(p. 275\)](#)

Determines whether Amazon Lex will use Amazon Comprehend to detect the sentiment of user utterances.

Type: [SentimentAnalysisSettings \(p. 856\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeBotLocale

Service: Amazon Lex Model Building V2

Describes the settings that a bot has for a specific locale.

### Request Syntax

```
GET /bots/botId/botversions/botVersion/botlocales/localeId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 279)

The identifier of the bot associated with the locale.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 279)

The identifier of the version of the bot associated with the locale.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 279)

The unique identifier of the locale to describe. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botLocaleHistoryEvents": [
        {
            "event": "string",
            "eventDate": number
        }
    ],
    "botLocaleStatus": "string",
    "botVersion": "string",
```

```
"creationDateTime": number,
"description": string,
"failureReasons": [ string ],
"intentsCount": number,
"lastBuildSubmittedDateTime": number,
"lastUpdatedDateTime": number,
"localeId": string,
"localeName": string,
"nluIntentConfidenceThreshold": number,
"recommendedActions": [ string ],
"slotTypesCount": number,
"voiceSettings": {
    "engine": string,
    "voiceId": string
}
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 279\)](#)

The identifier of the bot associated with the locale.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botLocaleHistoryEvents \(p. 279\)](#)

History of changes, such as when a locale is used in an alias, that have taken place for the locale.

Type: Array of [BotLocaleHistoryEvent \(p. 754\)](#) objects

### [botLocaleStatus \(p. 279\)](#)

The status of the bot. If the status is Failed, the reasons for the failure are listed in the failureReasons field.

Type: String

Valid Values: Creating | Building | Built | ReadyExpressTesting | Failed | Deleting | NotBuilt | Importing | Processing

### [botVersion \(p. 279\)](#)

The identifier of the version of the bot associated with the locale.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### [creationDateTime \(p. 279\)](#)

The date and time that the locale was created.

Type: Timestamp

[description \(p. 279\)](#)

The description of the locale.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

[failureReasons \(p. 279\)](#)

if botLocaleStatus is Failed, Amazon Lex explains why it failed to build the bot.

Type: Array of strings

[intentsCount \(p. 279\)](#)

The number of intents defined for the locale.

Type: Integer

[lastBuildSubmittedDateTime \(p. 279\)](#)

The date and time that the locale was last submitted for building.

Type: Timestamp

[lastUpdatedDateTime \(p. 279\)](#)

The date and time that the locale was last updated.

Type: Timestamp

[localeId \(p. 279\)](#)

The unique identifier of the described locale.

Type: String

[localeName \(p. 279\)](#)

The name of the locale.

Type: String

[nluIntentConfidenceThreshold \(p. 279\)](#)

The confidence threshold where Amazon Lex inserts the AMAZON.FallbackIntent and AMAZON.KendraSearchIntent intents in the list of possible intents for an utterance.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 1.

[recommendedActions \(p. 279\)](#)

Recommended actions to take to resolve an error in the failureReasons field.

Type: Array of strings

[slotTypesCount \(p. 279\)](#)

The number of slot types defined for the locale.

Type: Integer

[voiceSettings \(p. 279\)](#)

The Amazon Polly voice Amazon Lex uses for voice interaction with the user.

Type: [VoiceSettings \(p. 890\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeBotRecommendation

Service: Amazon Lex Model Building V2

Provides metadata information about a bot recommendation. This information will enable you to get a description on the request inputs, to download associated transcripts after processing is complete, and to download intents and slot-types generated by the bot recommendation.

### Request Syntax

```
GET /bots/botId/botversions/botVersion/botlocales/localeId/  
botrecommendations/botRecommendationId HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 283)

The unique identifier of the bot associated with the bot recommendation.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botRecommendationId** (p. 283)

The identifier of the bot recommendation to describe.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 283)

The version of the bot associated with the bot recommendation.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 283)

The identifier of the language and locale of the bot recommendation to describe. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200  
Content-type: application/json
```

```
{
    "botId": "string",
    "botRecommendationId": "string",
    "botRecommendationResults": {
        "associatedTranscriptsUrl": "string",
        "botLocaleExportUrl": "string",
        "statistics": {
            "intents": {
                "discoveredIntentCount": number
            },
            "slotTypes": {
                "discoveredSlotTypeCount": number
            }
        }
    },
    "botRecommendationStatus": "string",
    "botVersion": "string",
    "creationDateTime": number,
    "encryptionSetting": {
        "associatedTranscriptsPassword": "string",
        "botLocaleExportPassword": "string",
        "kmsKeyArn": "string"
    },
    "failureReasons": [ "string" ],
    "lastUpdatedDateTime": number,
    "localeId": "string",
    "transcriptSourceSetting": {
        "s3BucketTranscriptSource": {
            "kmsKeyArn": "string",
            "pathFormat": {
                "objectPrefixes": [ "string" ]
            },
            "s3BucketName": "string",
            "transcriptFilter": {
                "lexTranscriptFilter": {
                    "dateRangeFilter": {
                        "endDateTime": number,
                        "startDateTime": number
                    }
                }
            },
            "transcriptFormat": "string"
        }
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId (p. 283)**

The identifier of the bot associated with the bot recommendation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botRecommendationId (p. 283)**

The identifier of the bot recommendation being described.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**[botRecommendationResults \(p. 283\)](#)**

The object representing the URL of the bot definition, the URL of the associated transcript and a statistical summary of the bot recommendation results.

Type: [BotRecommendationResults \(p. 760\)](#) object

**[botRecommendationStatus \(p. 283\)](#)**

The status of the bot recommendation. If the status is Failed, then the reasons for the failure are listed in the failureReasons field.

Type: String

Valid Values: Processing | Deleting | Deleted | Downloading | Updating | Available | Failed | Stopping | Stopped

**[botVersion \(p. 283\)](#)**

The version of the bot associated with the bot recommendation.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

**[creationDateTime \(p. 283\)](#)**

The date and time that the bot recommendation was created.

Type: Timestamp

**[encryptionSetting \(p. 283\)](#)**

The object representing the passwords that were used to encrypt the data related to the bot recommendation results, as well as the KMS key ARN used to encrypt the associated metadata.

Type: [EncryptionSetting \(p. 796\)](#) object

**[failureReasons \(p. 283\)](#)**

If botRecommendationStatus is Failed, Amazon Lex explains why.

Type: Array of strings

**[lastUpdatedDateTime \(p. 283\)](#)**

The date and time that the bot recommendation was last updated.

Type: Timestamp

**[localeId \(p. 283\)](#)**

The identifier of the language and locale of the bot recommendation to describe.

Type: String

**[transcriptSourceSetting \(p. 283\)](#)**

The object representing the Amazon S3 bucket containing the transcript, as well as the associated metadata.

Type: [TranscriptSourceSetting \(p. 888\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeBotVersion

Service: Amazon Lex Model Building V2

Provides metadata about a version of a bot.

### Request Syntax

```
GET /bots/botId/botversions/botVersion HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 287)

The identifier of the bot containing the version to return metadata for.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 287)

The version of the bot to return metadata for.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botName": "string",
  "botStatus": "string",
  "botVersion": "string",
  "creationDateTime": number,
  "dataPrivacy": {
    "childDirected": boolean
  },
  "description": "string",
  "failureReasons": [ "string" ],
  "idleSessionTTLInSeconds": number,
  "roleArn": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**botId (p. 287)**

The identifier of the bot that contains the version.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botName (p. 287)**

The name of the bot that contains the version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

**botStatus (p. 287)**

The current status of the bot. When the status is Available, the bot version is ready for use.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

**botVersion (p. 287)**

The version of the bot to describe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

**creationDateTime (p. 287)**

A timestamp of the date and time that the bot version was created.

Type: Timestamp

**dataPrivacy (p. 287)**

Data privacy settings for the bot version.

Type: [DataPrivacy \(p. 787\)](#) object

**description (p. 287)**

The description specified for the bot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**failureReasons (p. 287)**

If the botStatus is Failed, this contains a list of reasons that the version couldn't be built.

Type: Array of strings

### [idleSessionTTLInSeconds \(p. 287\)](#)

The number of seconds that a session with the bot remains active before it is discarded by Amazon Lex.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 86400.

### [roleArn \(p. 287\)](#)

The Amazon Resource Name (ARN) of an IAM role that has permission to access the bot version.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:iam::[0-9]{12}:role/.\*\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeCustomVocabularyMetadata

Service: Amazon Lex Model Building V2

Provides metadata information about a custom vocabulary.

### Request Syntax

```
GET /bots/botId/botversions/botVersion/botlocales/localeId/customvocabulary/DEFAULT/metadata HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 291)

The unique identifier of the bot that contains the custom vocabulary.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 291)

The bot version of the bot to return metadata for.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 291)

The locale to return the custom vocabulary information for. The locale must be en\_GB.

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "creationDateTime": number,
    "customVocabularyStatus": "string",
    "lastUpdatedDateTime": number,
    "localeId": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**botId (p. 291)**

The identifier of the bot that contains the custom vocabulary.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botVersion (p. 291)**

The version of the bot that contains the custom vocabulary to describe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

**creationDateTime (p. 291)**

The date and time that the custom vocabulary was created.

Type: Timestamp

**customVocabularyStatus (p. 291)**

The status of the custom vocabulary. If the status is Ready the custom vocabulary is ready to use.

Type: String

Valid Values: Ready | Deleting | Exporting | Importing | Creating

**lastUpdatedDateTime (p. 291)**

The date and time that the custom vocabulary was last updated.

Type: Timestamp

**localeId (p. 291)**

The locale that contains the custom vocabulary to describe.

Type: String

**Errors**

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

**InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeExport

Service: Amazon Lex Model Building V2

Gets information about a specific export.

### Request Syntax

```
GET /exports/exportId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **exportId** (p. 294)

The unique identifier of the export to describe.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "creationDateTime": number,
    "downloadUrl": "string",
    "exportId": "string",
    "exportStatus": "string",
    "failureReasons": [ "string" ],
    "fileFormat": "string",
    "lastUpdatedDateTime": number,
    "resourceSpecification": {
        "botExportSpecification": {
            "botId": "string",
            "botVersion": "string"
        },
        "botLocaleExportSpecification": {
            "botId": "string",
            "botVersion": "string",
            "localeId": "string"
        },
        "customVocabularyExportSpecification": {
            "botId": "string",
            "botVersion": "string",
            "localeId": "string"
        }
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [creationDateTime \(p. 294\)](#)

The date and time that the export was created.

Type: Timestamp

### [downloadUrl \(p. 294\)](#)

A pre-signed S3 URL that points to the bot or bot locale archive. The URL is only available for 5 minutes after calling the `DescribeExport` operation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

### [exportId \(p. 294\)](#)

The unique identifier of the described export.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [exportStatus \(p. 294\)](#)

The status of the export. When the status is `Complete` the export archive file is available for download.

Type: String

Valid Values: `InProgress` | `Completed` | `Failed` | `Deleting`

### [failureReasons \(p. 294\)](#)

If the `exportStatus` is `Failed`, contains one or more reasons why the export could not be completed.

Type: Array of strings

### [fileFormat \(p. 294\)](#)

The file format used in the files that describe the resource.

Type: String

Valid Values: `LexJson` | `TSV`

### [lastUpdatedDateTime \(p. 294\)](#)

The last date and time that the export was updated.

Type: Timestamp

### [resourceSpecification \(p. 294\)](#)

The bot, bot ID, and optional locale ID of the exported bot or bot locale.

Type: [ExportResourceSpecification \(p. 798\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeImport

Service: Amazon Lex Model Building V2

Gets information about a specific import.

### Request Syntax

```
GET /imports/importId HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **importId** (p. 297)

The unique identifier of the import to describe.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "creationDateTime": number,
    "failureReasons": [ "string" ],
    "importedResourceId": "string",
    "importedResourceName": "string",
    "importId": "string",
    "importStatus": "string",
    "lastUpdatedDateTime": number,
    "mergeStrategy": "string",
    "resourceSpecification": {
        "botImportSpecification": {
            "botName": "string",
            "botTags": {
                "string": "string"
            },
            "dataPrivacy": {
                "childDirected": boolean
            },
            "idleSessionTTLInSeconds": number,
            "roleArn": "string",
            "testBotAliasTags": {
                "string": "string"
            }
        },
        "botLocaleImportSpecification": {
            "botId": "string",
            "botVersion": "string",
            "localeId": "string",
        }
    }
}
```

```
        "nluIntentConfidenceThreshold": number,
        "voiceSettings": {
            "engine": "string",
            "voiceId": "string"
        }
    },
    "customVocabularyImportSpecification": {
        "botId": "string",
        "botVersion": "string",
        "localeId": "string"
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [creationDateTime \(p. 297\)](#)

The date and time that the import was created.

Type: Timestamp

### [failureReasons \(p. 297\)](#)

If the `importStatus` field is Failed, this provides one or more reasons for the failure.

Type: Array of strings

### [importedResourceId \(p. 297\)](#)

The unique identifier that Amazon Lex assigned to the resource created by the import.

Type: String

Length Constraints: Minimum length of 5. Maximum length of 10.

Pattern: `^([0-9a-zA-Z])+$`

### [importedResourceName \(p. 297\)](#)

The name of the imported resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^([0-9a-zA-Z][_-]?)+$`

### [importId \(p. 297\)](#)

The unique identifier of the described import.

Type: String

Length Constraints: Fixed length of 10.

Pattern: `^[0-9a-zA-Z]+$`

### [importStatus \(p. 297\)](#)

The status of the import process. When the status is Completed the resource is imported and ready for use.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

[lastUpdatedDateTime \(p. 297\)](#)

The date and time that the import was last updated.

Type: Timestamp

[mergeStrategy \(p. 297\)](#)

The strategy used when there was a name conflict between the imported resource and an existing resource. When the merge strategy is FailOnConflict existing resources are not overwritten and the import fails.

Type: String

Valid Values: Overwrite | FailOnConflict | Append

[resourceSpecification \(p. 297\)](#)

The specifications of the imported bot, bot locale, or custom vocabulary.

Type: [ImportResourceSpecification \(p. 813\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

**InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeIntent

Service: Amazon Lex Model Building V2

Returns metadata about an intent.

### Request Syntax

```
GET /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 301)

The identifier of the bot associated with the intent.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 301)

The version of the bot associated with the intent.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **intentId** (p. 301)

The identifier of the intent to describe.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **localeId** (p. 301)

The identifier of the language and locale of the intent to describe. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
```

```
Content-type: application/json
```

```
{  
    "botId": "string",
```

```
"botVersion": "string",
"creationDateTime": number,
"description": "string",
"dialogCodeHook": {
    "enabled": boolean
},
"fulfillmentCodeHook": {
    "active": boolean,
    "enabled": boolean,
    "fulfillmentUpdatesSpecification": {
        "active": boolean,
        "startResponse": {
            "allowInterrupt": boolean,
            "delayInSeconds": number,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "imageUrl": "string",
                                "subtitle": "string",
                                "title": "string"
                            },
                            "plainTextMessage": {
                                "value": "string"
                            },
                            "ssmlMessage": {
                                "value": "string"
                            }
                        }
                    ]
                }
            ],
            "timeoutInSeconds": number,
```

```
"updateResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "postFulfillmentStatusSpecification": {
        "failureConditional": {
            "active": boolean,
            "conditionalBranches": [
                {
                    "condition": {
                        "expressionString": "string"
                    },
                    "name": "string",
                    "nextStep": {
                        "dialogAction": {
```

```
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            }
        }
    ],
    "sessionAttributes": {
        "string" : "string"
    }
}
```

```

        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ],
        "defaultBranch": {
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",

```

```
"slots": {  
    "string" : {  
        "shape": "string",  
        "value": {  
            "interpretedValue": "string"  
        },  
        "values": [  
            "SlotValueOverride"  
        ]  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"response": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    },  
                    "ssmlMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        }
      ],
    }
  },
  "successNextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

```
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {

```

```
"message": {
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                }
            },
            "values": [

```

```
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"initialResponseSetting": {
    "codeHook": {
```

```
"active": boolean,
"enableCodeHookInvocation": boolean,
"invocationLabel": string,
"postCodeHookSpecification": {
    "failureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": string
                },
                "name": string,
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": string,
                        "suppressNextMessage": boolean,
                        "type": string
                    },
                    "intent": {
                        "name": string,
                        "slots": {
                            string : {
                                "shape": string,
                                "value": {
                                    "interpretedValue": string
                                },
                                "values": [
                                    "SlotValueOverride"
                                ]
                            }
                        }
                    }
                },
                "sessionAttributes": {
                    string : string
                }
            }
        ],
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": string
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": string,
                                    "value": string
                                }
                            ],
                            "imageUrl": string,
                            "subtitle": string,
                            "title": string
                        },
                        "plainTextMessage": {
                            "value": string
                        },
                        "ssmlMessage": {
                            "value": string
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": string
                            }
                        }
                    ]
                }
            ]
        }
    }
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                }
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [

```

```
{ "message": { "customPayload": { "value": "string" } }, "imageResponseCard": { "buttons": [ { "text": "string", "value": "string" } ], "imageUrl": "string", "subtitle": "string", "title": "string" }, "plainTextMessage": { "value": "string" }, "ssmlMessage": { "value": "string" } }, "variations": [ { "customPayload": { "value": "string" } }, "imageResponseCard": { "buttons": [ { "text": "string", "value": "string" } ], "imageUrl": "string", "subtitle": "string", "title": "string" }, "plainTextMessage": { "value": "string" }, "ssmlMessage": { "value": "string" } } ] }, "successConditional": { "active": boolean, "conditionalBranches": [ { "condition": { "expressionString": "string" } }, "name": "string", "nextStep": { "dialogAction": { "slotToElicit": "string", "suppressNextMessage": boolean, "type": "string" } }, "intent": { "name": "string" } ] } }
```

```
"slots": {
    "string" : {
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        ]
    ]
}
```

```
        }
      ]
    }
  ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ]
              }
            }
          ]
        }
      ]
    }
  }
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {

```

```
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
{
        "condition": {
            "expressionString": "string"
        },
        "name": "string",
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {
                "name": "string",
                "slots": {
                    "string" : {
                        "shape": "string",
                        "value": {
                            "interpretedValue": "string"
                        },
                        "values": [
                            "SlotValueOverride"
                        ]
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
{

```

```
"message": {
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    }
                },
                ...
            }
        }
    }
}
```

```
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "timeoutNextStep": {
```

```
"dialogAction": {  
    "$slotToElicit": "string",  
    "$suppressNextMessage": boolean,  
    "type": "string"  
},  
"intent": {  
    "name": "string",  
    "slots": {  
        "string" : {  
            "shape": "string",  
            "value": {  
                "interpretedValue": "string"  
            },  
            "values": [  
                "SlotValueOverride"  
            ]  
        }  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"timeoutResponse": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"conditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "imageUrl": "string",
                                "subtitle": "string",
                                "title": "string"
                            },
                            "plainTextMessage": {
                                "value": "string"
                            },
                            "text": "string"
                        }
                    }
                ]
            }
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "text": "string"
                }
            }
        ]
    }
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"initialResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"inputContexts": [
    {
        "name": "string"
    }
],
"intentClosingSetting": {
    "active": boolean,
```

```
"closingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "conditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                        "type": "string"
                    }
                }
            }
        ]
    }
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    }
}
```

```
        }
      ]
    }
  ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [

```

```
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
}
],
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
}
},
"intentConfirmationSetting": {
    "active": boolean,
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {
                "active": boolean,
                "conditionalBranches": [
                    {
                        "condition": {
                            "expressionString": "string"
                        },
                        "name": "string",
                        "nextStep": {
                            "dialogAction": {
                                "slotToElicit": "string",
                                "suppressNextMessage": boolean,
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

```
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {

```

```
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
{
        "condition": {
            "expressionString": "string"
        },
        "name": "string",
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {
                "name": "string",
                "slots": {
                    "string" : {
                        "shape": "string",
                        "value": {
                            "interpretedValue": "string"
                        },
                        "values": [
                            "SlotValueOverride"
                        ]
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        }
    }
],
"failureCondition": {
    "active": boolean,
    "failureType": "string"
}
}
]
```

```
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {
                "name": "string",
                "slots": {
                    "string" : {

```

```
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
    },
    "successNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "successResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    },
                    "plainTextMessage": {

```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
    {
        "message": {
            "customPayload": {
```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                }
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    }
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
},
"confirmationConditional": {
    "active": boolean,
    "conditionalBranches": [
        {

```

```
"condition": {  
    "expressionString": "string"  
},  
"name": "string",  
"nextStep": {  
    "dialogAction": {  
        "slotToElicit": "string",  
        "suppressNextMessage": boolean,  
        "type": "string"  
    },  
    "intent": {  
        "name": "string",  
        "slots": {  
            "string" : {  
                "shape": "string",  
                "value": {  
                    "interpretedValue": "string"  
                },  
                "values": [  
                    "SlotValueOverride"  
                ]  
            }  
        }  
    },  
    "sessionAttributes": {  
        "string" : "string"  
    }  
},  
"response": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ]  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    },  
                    "ssmlMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            }
        ]
    }
}
```

```
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"confirmationNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"confirmationResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "text": "string",
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"declinationConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            }
                        },
                        "values": [
                            "SlotValueOverride"
                        ]
                    }
                }
            }
        }
    ]
}
```

```
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
```

```
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
},
"declinationNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"declinationResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },

```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "text": "string"
                    }
                }
            ]
        }
    ]
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string": {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    }
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    }
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "failureResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
}
```

```
        }
      ],
    ],
  },
  "promptSpecification": {
    "allowInterrupt": boolean,
    "maxRetries": number,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          }
        ]
      }
    ],
    "messageSelectionStrategy": "string",
    "promptAttemptsSpecification": {
      "string": {
        "allowedInputTypes": {
          "allowAudioInput": boolean,
          "allowDTMFInput": boolean
        },
        ...
      }
    }
  }
},
```

```
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeTimeoutMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": string,
                "endCharacter": string,
                "endTimeTimeoutMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"intentId": string,
"intentName": string,
"kendraConfiguration": {
    "kendraIndex": string,
    "queryFilterString": string,
    "queryFilterStringEnabled": boolean
},
"lastUpdatedDateTime": number,
"localeId": string,
"outputContexts": [
    {
        "name": string,
        "timeToLiveInSeconds": number,
        "turnsToLive": number
    }
],
"parentIntentSignature": string,
"sampleUtterances": [
    {
        "utterance": string
    }
],
"slotPriorities": [
    {
        "priority": number,
        "slotId": string
    }
]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId (p. 301)**

The identifier of the bot associated with the intent.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botVersion (p. 301)**

The version of the bot associated with the intent.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

**creationDateTime (p. 301)**

A timestamp of the date and time that the intent was created.

Type: Timestamp

**description (p. 301)**

The description of the intent.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**dialogCodeHook (p. 301)**

The Lambda function called during each turn of a conversation with the intent.

Type: [DialogCodeHookSettings \(p. 792\)](#) object

**fulfillmentCodeHook (p. 301)**

The Lambda function called when the intent is complete and ready for fulfillment.

Type: [FulfillmentCodeHookSettings \(p. 804\)](#) object

**initialResponseSetting (p. 301)**

Type: [InitialResponseSetting \(p. 817\)](#) object

**inputContexts (p. 301)**

A list of contexts that must be active for the intent to be considered for sending to the user.

Type: Array of [InputContext \(p. 818\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

**intentClosingSetting (p. 301)**

The response that Amazon Lex sends to when the intent is closed.

Type: [IntentClosingSetting \(p. 819\)](#) object

**intentConfirmationSetting (p. 301)**

Prompts that Amazon Lex sends to the user to confirm completion of an intent.

Type: [IntentConfirmationSetting \(p. 820\)](#) object

**intentId (p. 301)**

The unique identifier assigned to the intent when it was created.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [intentName \(p. 301\)](#)

The name specified for the intent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

#### [kendraConfiguration \(p. 301\)](#)

Configuration information required to use the AMAZON.KendraSearchIntent intent.

Type: [KendraConfiguration \(p. 829\)](#) object

#### [lastUpdatedDateTime \(p. 301\)](#)

A timestamp of the date and time that the intent was last updated.

Type: Timestamp

#### [localeId \(p. 301\)](#)

The language and locale specified for the intent.

Type: String

#### [outputContexts \(p. 301\)](#)

A list of contexts that are activated when the intent is fulfilled.

Type: Array of [OutputContext \(p. 837\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

#### [parentIntentSignature \(p. 301\)](#)

The identifier of the built-in intent that this intent is derived from, if any.

Type: String

#### [sampleUtterances \(p. 301\)](#)

User utterances that trigger this intent.

Type: Array of [SampleUtterance \(p. 854\)](#) objects

#### [slotPriorities \(p. 301\)](#)

The list that determines the priority that slots should be elicited from the user.

Type: Array of [SlotPriority \(p. 862\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeResourcePolicy

Service: Amazon Lex Model Building V2

Gets the resource policy and policy revision for a bot or bot alias.

### Request Syntax

```
GET /policy/resourceArn/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### [resourceArn \(p. 361\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "policy": "string",
  "resourceArn": "string",
  "revisionId": "string"
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### [policy \(p. 361\)](#)

The JSON structure that contains the resource policy. For more information about the contents of a JSON policy document, see [IAM JSON policy reference](#).

Type: String

Length Constraints: Minimum length of 2.

#### [resourceArn \(p. 361\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

### [revisionId \(p. 361\)](#)

The current revision of the resource policy. Use the revision ID to make sure that you are updating the most current version of a resource policy when you add a policy statement to a resource, delete a resource, or update a resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeSlot

Service: Amazon Lex Model Building V2

Gets metadata information about a slot.

### Request Syntax

```
GET /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/slots/slotId/  
HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 363)

The identifier of the bot associated with the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 363)

The version of the bot associated with the slot.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **intentId** (p. 363)

The identifier of the intent that contains the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **localeId** (p. 363)

The identifier of the language and locale of the slot to describe. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

#### **slotId** (p. 363)

The unique identifier for the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "creationDateTime": number,
    "description": "string",
    "intentId": "string",
    "lastUpdatedDateTime": number,
    "localeId": "string",
    "multipleValuesSetting": {
        "allowMultipleValues": boolean
    },
    "obfuscationSetting": {
        "obfuscationSettingType": "string"
    },
    "slotId": "string",
    "slotName": "string",
    "slotTypeId": "string",
    "subSlotSetting": {
        "expression": "string",
        "slotSpecifications": {
            "string" : {
                "slotTypeId": "string",
                "valueElicitationSetting": {
                    "defaultValueSpecification": {
                        "defaultValueList": [
                            {
                                "defaultValue": "string"
                            }
                        ]
                    },
                    "promptSpecification": {
                        "allowInterrupt": boolean,
                        "maxRetries": number,
                        "messageGroups": [
                            {
                                "message": {
                                    "customPayload": {
                                        "value": "string"
                                    },
                                    "imageResponseCard": {
                                        "buttons": [
                                            {
                                                "text": "string",
                                                "value": "string"
                                            }
                                        ],
                                        "imageUrl": "string",
                                        "subtitle": "string",
                                        "title": "string"
                                    },
                                    "plainTextMessage": {
                                        "value": "string"
                                    },
                                    "ssmlMessage": {
                                        "value": "string"
                                    }
                                }
                            }
                        ]
                    }
                }
            }
        }
    }
}
```

```
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeoutMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeoutMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {

```

```
"message": {
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
],
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
{
    "message": {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
{

```

```
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
]
}
}
}
},
"valueElicitationSetting": {
    "defaultValueSpecification": {
        "defaultValueList": [
            {
                "defaultValue": "string"
            }
        ]
    },
    "promptSpecification": {
        "allowInterrupt": boolean,
        "maxRetries": number,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [

```

```
{  
    "customPayload": {  
        "value": "string"  
    },  
    "imageResponseCard": {  
        "buttons": [  
            {  
                "text": "string",  
                "value": "string"  
            }  
        ],  
        "imageUrl": "string",  
        "subtitle": "string",  
        "title": "string"  
    },  
    "plainTextMessage": {  
        "value": "string"  
    },  
    "ssmlMessage": {  
        "value": "string"  
    }  
}  
]  
],  
"messageSelectionStrategy": "string",  
"promptAttemptsSpecification": {  
    "string" : {  
        "allowedInputTypes": {  
            "allowAudioInput": boolean,  
            "allowDTMFInput": boolean  
        },  
        "allowInterrupt": boolean,  
        "audioAndDTMFInputSpecification": {  
            "audioSpecification": {  
                "endTimeoutMs": number,  
                "maxLengthMs": number  
            },  
            "dtmfSpecification": {  
                "deletionCharacter": "string",  
                "endCharacter": "string",  
                "endTimeoutMs": number,  
                "maxLength": number  
            },  
            "startTimeoutMs": number  
        },  
        "textInputSpecification": {  
            "startTimeoutMs": number  
        }  
    }  
},  
"sampleUtterances": [  
    {  
        "utterance": "string"  
    }  
],  
"slotCaptureSetting": {  
    "captureConditional": {  
        "active": boolean,  
        "conditionalBranches": [  
            {  
                "condition": {  
                    "expressionString": "string"  
                },  
                "name": "string",  
                "value": "string"  
            }  
        ]  
    }  
}
```

```
"nextStep": {  
    "dialogAction": {  
        "slotToElicit": "string",  
        "suppressNextMessage": boolean,  
        "type": "string"  
    },  
    "intent": {  
        "name": "string",  
        "slots": {  
            "string" : {  
                "shape": "string",  
                "value": {  
                    "interpretedValue": "string"  
                },  
                "values": [  
                    "SlotValueOverride"  
                ]  
            }  
        }  
    },  
    "sessionAttributes": {  
        "string" : "string"  
    }  
},  
"response": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    },  
                    "ssmlMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {

```

```
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"captureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"captureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"codeHook": {
    "active": boolean,
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string",
    "postCodeHookSpecification": {
        "failureConditional": {
            "active": boolean,
            "conditionalBranches": [
                {
                    "condition": {
                        "expressionString": "string"
                    },
                    "name": "string",
                    "nextStep": {
                        "dialogAction": {
                            "slotToElicit": "string",
                            "suppressNextMessage": boolean,
                            "type": "string"
                        },
                        "intent": {
                            "name": "string",
                            "slots": {
                                "string" : {
                                    "shape": "string",
                                    "value": {
                                        "interpretedValue": "string"
                                    }
                                },
                                "values": [

```

```
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"defaultBranch": {
    "nextStep": {
```

```
"dialogAction": {  
    "slotToElicit": "string",  
    "suppressNextMessage": boolean,  
    "type": "string"  
},  
"intent": {  
    "name": "string",  
    "slots": {  
        "string" : {  
            "shape": "string",  
            "value": {  
                "interpretedValue": "string"  
            },  
            "values": [  
                "SlotValueOverride"  
            ]  
        }  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"response": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
},
],
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {

```

```
        "text": "string",
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
```

```
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "successNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "value": "string"
        }
    }
}
```

```
"slots": {  
    "string" : {  
        "shape": "string",  
        "value": {  
            "interpretedValue": "string"  
        },  
        "values": [  
            "SlotValueOverride"  
        ]  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"successResponse": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    },  
                    "ssmlMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

```
        }
    ],
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": "string"
                            }
                        },
                        {
                            "customPayload": {
                                "value": "string"
                            }
                        }
                    ]
                }
            ]
        }
    ]
}
```

```
    "imageResponseCard": {
      "buttons": [
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string": {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string": "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          }
        }
      }
    ]
  }
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
    {

```

```
"message": {
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "type": "string"
                }
            }
        }
    ]
}
```

```
    "suppressNextMessage": boolean,
    "type": "string"
},
"intent": {
    "name": "string",
    "slots": {
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },

```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"slotConstraint": "string",
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            },
            "variations": [

```

```
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
]
}
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ]  
    } }  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 364\)](#)

The identifier of the bot associated with the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 364\)](#)

The version of the bot associated with the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### [creationDateTime \(p. 364\)](#)

A timestamp of the date and time that the slot was created.

Type: Timestamp

### [description \(p. 364\)](#)

The description specified for the slot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

### [intentId \(p. 364\)](#)

The identifier of the intent associated with the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [lastUpdatedDateTime \(p. 364\)](#)

A timestamp of the date and time that the slot was last updated.

Type: Timestamp

### [localeId \(p. 364\)](#)

The language and locale specified for the slot.

Type: String

**multipleValuesSetting (p. 364)**

Indicates whether the slot accepts multiple values in a single utterance.

If the `multipleValuesSetting` is not set, the default value is `false`.

Type: [MultipleValuesSetting \(p. 834\)](#) object

**obfuscationSetting (p. 364)**

Whether slot values are shown in Amazon CloudWatch logs. If the value is `None`, the actual value of the slot is shown in logs.

Type: [ObfuscationSetting \(p. 836\)](#) object

**slotId (p. 364)**

The unique identifier generated for the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: `^[0-9a-zA-Z]+$`

**slotName (p. 364)**

The name specified for the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^([0-9a-zA-Z][_-]?)+$`

**slotTypeId (p. 364)**

The identifier of the slot type that determines the values entered into the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern: `^(AMAZON\.)[a-zA-Z_]+?|[0-9a-zA-Z]+)$`

**subSlotSetting (p. 364)**

Specifications for the constituent sub slots and the expression for the composite slot.

Type: [SubSlotSetting \(p. 881\)](#) object

**valueElicitationSetting (p. 364)**

Prompts that Amazon Lex uses to elicit a value for the slot.

Type: [SlotValueElicitationSetting \(p. 873\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### InternalServerError

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeSlotType

Service: Amazon Lex Model Building V2

Gets metadata information about a slot type.

### Request Syntax

```
GET /bots/botId/botversions/botVersion/botlocales/localeId/slottypes/slotTypeId/ HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 394)

The identifier of the bot associated with the slot type.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 394)

The version of the bot associated with the slot type.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 394)

The identifier of the language and locale of the slot type to describe. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

#### **slotTypeId** (p. 394)

The identifier of the slot type.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json
{
}
```

```
"botId": "string",
"botVersion": "string",
"compositeSlotTypeSetting": {
    "subSlots": [
        {
            "name": "string",
            "slotTypeId": "string"
        }
    ]
},
"creationDateTime": number,
"description": "string",
"externalSourceSetting": {
    "grammarSlotTypeSetting": {
        "source": {
            "kmsKeyArn": "string",
            "s3BucketName": "string",
            "s3ObjectKey": "string"
        }
    }
},
"lastUpdatedDateTime": number,
"localeId": "string",
"parentSlotTypeSignature": "string",
"slotTypeId": "string",
"slotTypeName": "string",
"slotTypeValues": [
    {
        "sampleValue": {
            "value": "string"
        },
        "synonyms": [
            {
                "value": "string"
            }
        ]
    }
],
"valueSelectionSetting": {
    "advancedRecognitionSetting": {
        "audioRecognitionStrategy": "string"
    },
    "regexFilter": {
        "pattern": "string"
    },
    "resolutionStrategy": "string"
}
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId (p. 394)**

The identifier of the bot associated with the slot type.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

[botVersion \(p. 394\)](#)

The version of the bot associated with the slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT | [0-9]+)\$

[compositeSlotTypeSetting \(p. 394\)](#)

Specifications for a composite slot type.

Type: [CompositeSlotTypeSetting \(p. 777\)](#) object

[creationDateTime \(p. 394\)](#)

A timestamp of the date and time that the slot type was created.

Type: Timestamp

[description \(p. 394\)](#)

The description specified for the slot type.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

[externalSourceSetting \(p. 394\)](#)

Provides information about the external source of the slot type's definition.

Type: [ExternalSourceSetting \(p. 802\)](#) object

[lastUpdatedDateTime \(p. 394\)](#)

A timestamp of the date and time that the slot type was last updated.

Type: Timestamp

[localeId \(p. 394\)](#)

The language and locale specified for the slot type.

Type: String

[parentSlotTypeSignature \(p. 394\)](#)

The built in slot type used as a parent to this slot type.

Type: String

[slotTypeId \(p. 394\)](#)

The unique identifier for the slot type.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

[slotTypeName \(p. 394\)](#)

The name specified for the slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

#### [slotTypeValues \(p. 394\)](#)

The values that the slot type can take. Includes any synonyms for the slot type values.

Type: Array of [SlotTypeValue \(p. 871\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10000 items.

#### [valueSelectionSetting \(p. 394\)](#)

The strategy that Amazon Lex uses to choose a value from a list of possible values.

Type: [SlotValueSelectionSetting \(p. 877\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListAggregatedUtterances

Service: Amazon Lex Model Building V2

Provides a list of utterances that users have sent to the bot.

Utterances are aggregated by the text of the utterance. For example, all instances where customers used the phrase "I want to order pizza" are aggregated into the same line in the response.

You can see both detected utterances and missed utterances. A detected utterance is where the bot properly recognized the utterance and activated the associated intent. A missed utterance was not recognized by the bot and didn't activate an intent.

Utterances can be aggregated for a bot alias or for a bot version, but not both at the same time.

Utterances statistics are not generated under the following conditions:

- The `childDirected` field was set to true when the bot was created.
- You are using slot obfuscation with one or more slots.
- You opted out of participating in improving Amazon Lex.

## Request Syntax

```
POST /bots/botId/aggregatedutterances/ HTTP/1.1
Content-type: application/json

{
  "aggregationDuration": {
    "relativeAggregationDuration": {
      "timeDimension": "string",
      "timeValue": number
    }
  },
  "botAliasId": "string",
  "botVersion": "string",
  "filters": [
    {
      "name": "string",
      "operator": "string",
      "values": [ "string" ]
    }
  ],
  "localeId": "string",
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

## URI Request Parameters

The request uses the following URI parameters.

### **botId** (p. 399)

The unique identifier of the bot associated with this request.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### **aggregationDuration (p. 399)**

The time window for aggregating the utterance information. You can specify a time between one hour and two weeks.

Type: [UtteranceAggregationDuration \(p. 889\)](#) object

Required: Yes

### **botAliasId (p. 399)**

The identifier of the bot alias associated with this request. If you specify the bot alias, you can't specify the bot version.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

Required: No

### **botVersion (p. 399)**

The identifier of the bot version associated with this request. If you specify the bot version, you can't specify the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: No

### **filters (p. 399)**

Provides the specification of a filter used to limit the utterances in the response to only those that match the filter specification. You can only specify one filter and one string to filter on.

Type: Array of [AggregatedUtterancesFilter \(p. 733\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

### **localeId (p. 399)**

The identifier of the language and locale where the utterances were collected. For more information, see [Supported languages](#).

Type: String

Required: Yes

### [maxResults \(p. 399\)](#)

The maximum number of utterances to return in each page of results. If there are fewer results than the maximum page size, only the actual number of results are returned. If you don't specify the maxResults parameter, 1,000 results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 399\)](#)

If the response from the `ListAggregatedUtterances` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response. Use that token in the `nextToken` parameter to return the next page of results.

Type: String

Required: No

### [sortBy \(p. 399\)](#)

Specifies sorting parameters for the list of utterances. You can sort by the hit count, the missed count, or the number of distinct sessions the utterance appeared in.

Type: [AggregatedUtterancesSortBy \(p. 734\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "aggregatedUtterancesSummaries": [
        {
            "containsDataFromDeletedResources": boolean,
            "hitCount": number,
            "missedCount": number,
            "utterance": "string",
            "utteranceFirstRecordedInAggregationDuration": number,
            "utteranceLastRecordedInAggregationDuration": number
        }
    ],
    "aggregationDuration": {
        "relativeAggregationDuration": {
            "timeDimension": "string",
            "timeValue": number
        }
    },
    "aggregationLastRefreshedDateTime": number,
    "aggregationWindowEndTime": number,
    "aggregationWindowStartTime": number,
    "botAliasId": "string",
    "botId": "string",
    "botVersion": "string",
    "localeId": "string",
    "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [aggregatedUtterancesSummaries \(p. 401\)](#)

Summaries of the aggregated utterance data. Each response contains information about the number of times that the utterance was seen during the time period, whether it was detected or missed, and when it was seen during the time period.

Type: Array of [AggregatedUtterancesSummary \(p. 735\)](#) objects

### [aggregationDuration \(p. 401\)](#)

The time period used to aggregate the utterance data.

Type: [UtteranceAggregationDuration \(p. 889\)](#) object

### [aggregationLastRefreshedDateTime \(p. 401\)](#)

The last date and time that the aggregated data was collected. The time period depends on the length of the aggregation window.

- **Hours** - for 1 hour time window, every half hour; otherwise every hour.
- **Days** - every 6 hours
- **Weeks** - for a one week time window, every 12 hours; otherwise, every day

Type: Timestamp

### [aggregationWindowEndTime \(p. 401\)](#)

The date and time that the aggregation window ends. Only data collected between the start time and the end time are returned in the results.

Type: Timestamp

### [aggregationWindowStartTime \(p. 401\)](#)

The date and time that the aggregation window begins. Only data collected after this time is returned in the results.

Type: Timestamp

### [botAliasId \(p. 401\)](#)

The identifier of the bot alias that contains the utterances. If you specified the bot version, the bot alias ID isn't returned.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

### [botId \(p. 401\)](#)

The identifier of the bot that contains the utterances.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 401\)](#)

The identifier of the bot version that contains the utterances. If you specified the bot alias, the bot version isn't returned.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### [localeId \(p. 401\)](#)

The identifier of the language and locale that the utterances are in.

Type: String

### [nextToken \(p. 401\)](#)

A token that indicates whether there are more results to return in a response to the `ListAggregatedUtterances` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `ListAggregatedUtterances` operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListBotAliases

Service: Amazon Lex Model Building V2

Gets a list of aliases for the specified bot.

### Request Syntax

```
POST /bots/botId/botaliases/ HTTP/1.1
Content-type: application/json

{
    "maxResults": number,
    "nextToken": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId (p. 405)**

The identifier of the bot to list aliases for.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **maxResults (p. 405)**

The maximum number of aliases to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

#### **nextToken (p. 405)**

If the response from the ListBotAliases operation contains more results than specified in the maxResults parameter, a token is returned in the response. Use that token in the nextToken parameter to return the next page of results.

Type: String

Required: No

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{  
    "botAliasSummaries": [  
        {  
            "botAliasId": "string",  
            "botAliasName": "string",  
            "botAliasStatus": "string",  
            "botVersion": "string",  
            "creationDateTime": number,  
            "description": "string",  
            "lastUpdatedDateTime": number  
        }  
    ],  
    "botId": "string",  
    "nextToken": "string"  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botAliasSummaries (p. 405)**

Summary information for the bot aliases that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter of the request. If there are more aliases available, the nextToken field contains a token to get the next page of results.

Type: Array of [BotAliasSummary \(p. 746\)](#) objects

### **botId (p. 405)**

The identifier of the bot associated with the aliases.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **nextToken (p. 405)**

A token that indicates whether there are more results to return in a response to the ListBotAliases operation. If the nextToken field is present, you send the contents as the nextToken parameter of a ListBotAliases operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListBotLocales

Service: Amazon Lex Model Building V2

Gets a list of locales for the specified bot.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/ HTTP/1.1
Content-type: application/json

{
  "filters": [
    {
      "name": "string",
      "operator": "string",
      "values": [ "string" ]
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 408)

The identifier of the bot to list locales for.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 408)

The version of the bot to list locales for.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **filters** (p. 408)

Provides the specification for a filter used to limit the response to only those locales that match the filter specification. You can only specify one filter and one value to filter on.

Type: Array of [BotLocaleFilter](#) (p. 753) objects

Array Members: Fixed number of 1 item.

Required: No

#### [maxResults \(p. 408\)](#)

The maximum number of aliases to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

#### [nextToken \(p. 408\)](#)

If the response from the `ListBotLocales` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response. Use that token as the `nextToken` parameter to return the next page of results.

Type: String

Required: No

#### [sortBy \(p. 408\)](#)

Specifies sorting parameters for the list of locales. You can sort by locale name in ascending or descending order.

Type: [BotLocaleSortBy \(p. 757\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botLocaleSummaries": [
        {
            "botLocaleStatus": "string",
            "description": "string",
            "lastBuildSubmittedDateTime": number,
            "lastUpdatedDateTime": number,
            "localeId": "string",
            "localeName": "string"
        }
    ],
    "botVersion": "string",
    "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### [botId \(p. 409\)](#)

The identifier of the bot to list locales for.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [botLocaleSummaries \(p. 409\)](#)

Summary information for the locales that meet the filter criteria specified in the request. The length of the list is specified in the `maxResults` parameter of the request. If there are more locales available, the `nextToken` field contains a token to get the next page of results.

Type: Array of [BotLocaleSummary \(p. 758\)](#) objects

#### [botVersion \(p. 409\)](#)

The version of the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

#### [nextToken \(p. 409\)](#)

A token that indicates whether there are more results to return in a response to the `ListBotLocales` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `ListBotLocales` operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListBotRecommendations

Service: Amazon Lex Model Building V2

Get a list of bot recommendations that meet the specified criteria.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/botrecommendations/ HTTP/1.1
Content-type: application/json

{
    "maxResults": number,
    "nextToken": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 412)

The unique identifier of the bot that contains the bot recommendation list.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 412)

The version of the bot that contains the bot recommendation list.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 412)

The identifier of the language and locale of the bot recommendation list.

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **maxResults** (p. 412)

The maximum number of bot recommendations to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 412\)](#)

If the response from the ListBotRecommendation operation contains more results than specified in the maxResults parameter, a token is returned in the response. Use that token in the nextToken parameter to return the next page of results.

Type: String

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botRecommendationSummaries": [
    {
      "botRecommendationId": "string",
      "botRecommendationStatus": "string",
      "creationDateTime": number,
      "lastUpdatedDateTime": number
    }
  ],
  "botVersion": "string",
  "localeId": "string",
  "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 413\)](#)

The unique identifier of the bot that contains the bot recommendation list.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botRecommendationSummaries \(p. 413\)](#)

Summary information for the bot recommendations that meet the filter specified in this request. The length of the list is specified in the maxResults parameter of the request. If there are more bot recommendations available, the nextToken field contains a token to get the next page of results.

Type: Array of [BotRecommendationSummary \(p. 762\)](#) objects

### [botVersion \(p. 413\)](#)

The version of the bot that contains the bot recommendation list.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### [localeId \(p. 413\)](#)

The identifier of the language and locale of the bot recommendation list.

Type: String

### [nextToken \(p. 413\)](#)

A token that indicates whether there are more results to return in a response to the ListBotRecommendations operation. If the nextToken field is present, you send the contents as the nextToken parameter of a ListBotRecommendations operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## ListBots

Service: Amazon Lex Model Building V2

Gets a list of available bots.

### Request Syntax

```
POST /bots/ HTTP/1.1
Content-type: application/json

{
  "filters": [
    {
      "name": "string",
      "operator": "string",
      "values": [ "string" ]
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

### URI Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### [filters \(p. 416\)](#)

Provides the specification of a filter used to limit the bots in the response to only those that match the filter specification. You can only specify one filter and one string to filter on.

Type: Array of [BotFilter \(p. 749\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

#### [maxResults \(p. 416\)](#)

The maximum number of bots to return in each page of results. If there are fewer results than the maximum page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

#### [nextToken \(p. 416\)](#)

If the response from the `ListBots` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response.

Use the returned token in the nextToken parameter of a ListBots request to return the next page of results. For a complete set of results, call the ListBots operation until the nextToken returned in the response is null.

Type: String

Required: No

#### [sortBy \(p. 416\)](#)

Specifies sorting parameters for the list of bots. You can specify that the list be sorted by bot name in ascending or descending order.

Type: [BotSortBy \(p. 763\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botSummaries": [
        {
            "botId": "string",
            "botName": "string",
            "botStatus": "string",
            "description": "string",
            "lastUpdatedDateTime": number,
            "latestBotVersion": "string"
        }
    ],
    "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botSummaries \(p. 417\)](#)

Summary information for the bots that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter of the request. If there are more bots available, the nextToken field contains a token to the next page of results.

Type: Array of [BotSummary \(p. 764\)](#) objects

### [nextToken \(p. 417\)](#)

A token that indicates whether there are more results to return in a response to the ListBots operation. If the nextToken field is present, you send the contents as the nextToken parameter of a ListBots operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListBotVersions

Service: Amazon Lex Model Building V2

Gets information about all of the versions of a bot.

The `ListBotVersions` operation returns a summary of each version of a bot. For example, if a bot has three numbered versions, the `ListBotVersions` operation returns four summaries, one for each numbered version and one for the DRAFT version.

The `ListBotVersions` operation always returns at least one version, the DRAFT version.

### Request Syntax

```
POST /bots/botId/botversions/ HTTP/1.1
Content-type: application/json

{
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 419)

The identifier of the bot to list versions for.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **maxResults** (p. 419)

The maximum number of versions to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

#### **nextToken** (p. 419)

If the response to the `ListBotVersion` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response. Use that token in the `nextToken` parameter to return the next page of results.

Type: String

Required: No

#### [sortBy \(p. 419\)](#)

Specifies sorting parameters for the list of versions. You can specify that the list be sorted by version name in either ascending or descending order.

Type: [BotVersionSortBy \(p. 767\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botVersionSummaries": [
    {
      "botName": "string",
      "botStatus": "string",
      "botVersion": "string",
      "creationDateTime": number,
      "description": "string"
    }
  ],
  "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### [botId \(p. 420\)](#)

The identifier of the bot to list versions for.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [botVersionSummaries \(p. 420\)](#)

Summary information for the bot versions that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter of the request. If there are more versions available, the nextToken field contains a token to get the next page of results.

Type: Array of [BotVersionSummary \(p. 768\)](#) objects

#### [nextToken \(p. 420\)](#)

A token that indicates whether there are more results to return in a response to the ListBotVersions operation. If the nextToken field is present, you send the contents as the nextToken parameter of a ListBotAliases operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListBuiltInIntents

Service: Amazon Lex Model Building V2

Gets a list of built-in intents provided by Amazon Lex that you can use in your bot.

To use a built-in intent as a the base for your own intent, include the built-in intent signature in the `parentIntentSignature` parameter when you call the `CreateIntent` operation. For more information, see [CreateIntent](#).

### Request Syntax

```
POST /builtins/locales/localeId/intents/ HTTP/1.1
Content-type: application/json

{
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **localeId (p. 422)**

The identifier of the language and locale of the intents to list. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **maxResults (p. 422)**

The maximum number of built-in intents to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Fixed value of 20.

Required: No

#### **nextToken (p. 422)**

If the response from the `ListBuiltInIntents` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response. Use that token in the `nextToken` parameter to return the next page of results.

Type: String

Required: No

### **sortBy (p. 422)**

Specifies sorting parameters for the list of built-in intents. You can specify that the list be sorted by the built-in intent signature in either ascending or descending order.

Type: [BuiltInIntentSortBy \(p. 770\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "builtInIntentSummaries": [
        {
            "description": "string",
            "intentSignature": "string"
        }
    ],
    "localeId": "string",
    "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **[builtInIntentSummaries \(p. 423\)](#)**

Summary information for the built-in intents that meet the filter criteria specified in the request. The length of the list is specified in the `maxResults` parameter of the request. If there are more intents available, the `nextToken` field contains a token to get the next page of results.

Type: Array of [BuiltInIntentSummary \(p. 771\)](#) objects

### **[localeId \(p. 423\)](#)**

The language and locale of the intents in the list.

Type: String

### **[nextToken \(p. 423\)](#)**

A token that indicates whether there are more results to return in a response to the `ListBuiltInIntents` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `ListBotAliases` operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **[InternalServerException](#)**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListBuiltInSlotTypes

Service: Amazon Lex Model Building V2

Gets a list of built-in slot types that meet the specified criteria.

### Request Syntax

```
POST /builtins/locales/localeId/slottypes/ HTTP/1.1
Content-type: application/json

{
    "maxResults": number,
    "nextToken": "string",
    "sortBy": {
        "attribute": "string",
        "order": "string"
    }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **localeId** (p. 425)

The identifier of the language and locale of the slot types to list. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **maxResults** (p. 425)

The maximum number of built-in slot types to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Fixed value of 20.

Required: No

#### **nextToken** (p. 425)

If the response from the ListBuiltInSlotTypes operation contains more results than specified in the maxResults parameter, a token is returned in the response. Use that token in the nextToken parameter to return the next page of results.

Type: String

Required: No

#### **sortBy** (p. 425)

Determines the sort order for the response from the ListBuiltInSlotTypes operation. You can choose to sort by the slot type signature in either ascending or descending order.

Type: [BuiltInSlotTypeSortBy](#) (p. 772) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "builtInSlotTypeSummaries": [
    {
      "description": "string",
      "slotTypeSignature": "string"
    }
  ],
  "localeId": "string",
  "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **builtInSlotTypeSummaries** (p. 426)

Summary information for the built-in slot types that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter of the request. If there are more slot types available, the nextToken field contains a token to get the next page of results.

Type: Array of [BuiltInSlotTypeSummary](#) (p. 773) objects

### **localeId** (p. 426)

The language and locale of the slot types in the list.

Type: String

### **nextToken** (p. 426)

A token that indicates whether there are more results to return in a response to the `ListBuiltInSlotTypes` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `LIstBuiltInSlotTypes` operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 936).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListCustomVocabularyItems

Service: Amazon Lex Model Building V2

Paginated list of custom vocabulary items for a given bot locale's custom vocabulary.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/customvocabulary/DEFAULT/list
HTTP/1.1
Content-type: application/json

{
    "maxResults": number,
    "nextToken": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 428)

The identifier of the version of the bot associated with this custom vocabulary.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 428)

The bot version of the bot to the list custom vocabulary request.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 428)

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see Supported languages (<https://docs.aws.amazon.com/lexv2/latest/dg/how-languages.html>).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **maxResults** (p. 428)

The maximum number of items returned by the list operation.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 428\)](#)

The nextToken identifier to the list custom vocabulary request.

Type: String

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botVersion": "string",
  "customVocabularyItems": [
    {
      "displayAs": "string",
      "itemId": "string",
      "phrase": "string",
      "weight": number
    }
  ],
  "localeId": "string",
  "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 429\)](#)

The identifier of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 429\)](#)

The identifier of the version of the bot associated with this custom vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### [customVocabularyItems \(p. 429\)](#)

The custom vocabulary items from the list custom vocabulary response.

Type: Array of [CustomVocabularyItem \(p. 786\)](#) objects

### [localeId \(p. 429\)](#)

The identifier of the language and locale where this custom vocabulary is used. The string must match one of the supported locales. For more information, see [Supported Languages](#).

Type: String

#### **nextToken (p. 429)**

The nextToken identifier to the list custom vocabulary response.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListExports

Service: Amazon Lex Model Building V2

Lists the exports for a bot, bot locale, or custom vocabulary. Exports are kept in the list for 7 days.

### Request Syntax

```
POST /exports/ HTTP/1.1
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "filters": [
        {
            "name": "string",
            "operator": "string",
            "values": [ "string" ]
        }
    ],
    "localeId": "string",
    "maxResults": number,
    "nextToken": "string",
    "sortBy": {
        "attribute": "string",
        "order": "string"
    }
}
```

### URI Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### **botId** (p. 431)

The unique identifier that Amazon Lex assigned to the bot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **botVersion** (p. 431)

The version of the bot to list exports for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: No

### [filters \(p. 431\)](#)

Provides the specification of a filter used to limit the exports in the response to only those that match the filter specification. You can only specify one filter and one string to filter on.

Type: Array of [ExportFilter \(p. 797\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

### [localeId \(p. 431\)](#)

Specifies the resources that should be exported. If you don't specify a resource type in the filters parameter, both bot locales and custom vocabularies are exported.

Type: String

Required: No

### [maxResults \(p. 431\)](#)

The maximum number of exports to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 431\)](#)

If the response from the `ListExports` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response.

Use the returned token in the `nextToken` parameter of a `ListExports` request to return the next page of results. For a complete set of results, call the `ListExports` operation until the `nextToken` returned in the response is null.

Type: String

Required: No

### [sortBy \(p. 431\)](#)

Determines the field that the list of exports is sorted by. You can sort by the `LastUpdatedDateTime` field in ascending or descending order.

Type: [ExportSortBy \(p. 799\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botVersion": "string",
  "exportSummaries": [
    {
      "status": "string"
    }
  ]
}
```

```
        "creationDateTime": number,
        "exportId": "string",
        "exportStatus": "string",
        "fileFormat": "string",
        "lastUpdatedDateTime": number,
        "resourceSpecification": {
            "botExportSpecification": {
                "botId": "string",
                "botVersion": "string"
            },
            "botLocaleExportSpecification": {
                "botId": "string",
                "botVersion": "string",
                "localeId": "string"
            },
            "customVocabularyExportSpecification": {
                "botId": "string",
                "botVersion": "string",
                "localeId": "string"
            }
        }
    ],
    "localeId": "string",
    "nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId** ([p. 432](#))

The unique identifier assigned to the bot by Amazon Lex.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion** ([p. 432](#))

The version of the bot that was exported.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### **exportSummaries** ([p. 432](#))

Summary information for the exports that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter. If there are more exports available, the nextToken field contains a token to get the next page of results.

Type: Array of [ExportSummary](#) ([p. 800](#)) objects

### **localeId** ([p. 432](#))

The locale specified in the request.

Type: String

#### [nextToken \(p. 432\)](#)

A token that indicates whether there are more results to return in a response to the `ListExports` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `ListExports` operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListImports

Service: Amazon Lex Model Building V2

Lists the imports for a bot, bot locale, or custom vocabulary. Imports are kept in the list for 7 days.

### Request Syntax

```
POST /imports/ HTTP/1.1
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "filters": [
        {
            "name": "string",
            "operator": "string",
            "values": [ "string" ]
        }
    ],
    "localeId": "string",
    "maxResults": number,
    "nextToken": "string",
    "sortBy": {
        "attribute": "string",
        "order": "string"
    }
}
```

### URI Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### **botId** (p. 435)

The unique identifier that Amazon Lex assigned to the bot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **botVersion** (p. 435)

The version of the bot to list imports for.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: No

### [filters \(p. 435\)](#)

Provides the specification of a filter used to limit the bots in the response to only those that match the filter specification. You can only specify one filter and one string to filter on.

Type: Array of [ImportFilter \(p. 812\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

### [localeId \(p. 435\)](#)

Specifies the locale that should be present in the list. If you don't specify a resource type in the filters parameter, the list contains both bot locales and custom vocabularies.

Type: String

Required: No

### [maxResults \(p. 435\)](#)

The maximum number of imports to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 435\)](#)

If the response from the `ListImports` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response.

Use the returned token in the `nextToken` parameter of a `ListImports` request to return the next page of results. For a complete set of results, call the `ListImports` operation until the `nextToken` returned in the response is null.

Type: String

Required: No

### [sortBy \(p. 435\)](#)

Determines the field that the list of imports is sorted by. You can sort by the `LastUpdatedDateTime` field in ascending or descending order.

Type: [ImportSortBy \(p. 814\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "importSummaries": [
        {
            "creationDateTime": number,
```

```
        "importedResourceId": "string",
        "importedResourceName": "string",
        "importedResourceType": "string",
        "importId": "string",
        "importStatus": "string",
        "lastUpdatedDateTime": number,
        "mergeStrategy": "string"
    }
],
"localeId": "string",
"nextToken": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId** (p. 436)

The unique identifier assigned by Amazon Lex to the bot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion** (p. 436)

The version of the bot that was imported. It will always be DRAFT.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **importSummaries** (p. 436)

Summary information for the imports that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter. If there are more imports available, the nextToken field contains a token to get the next page of results.

Type: Array of [ImportSummary](#) (p. 815) objects

### **localeId** (p. 436)

The locale specified in the request.

Type: String

### **nextToken** (p. 436)

A token that indicates whether there are more results to return in a response to the `ListImports` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `ListImports` operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 936).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListIntents

Service: Amazon Lex Model Building V2

Get a list of intents that meet the specified criteria.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/intents/ HTTP/1.1
Content-type: application/json

{
  "filters": [
    {
      "name": "string",
      "operator": "string",
      "values": [ "string" ]
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 439)

The unique identifier of the bot that contains the intent.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 439)

The version of the bot that contains the intent.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 439)

The identifier of the language and locale of the intents to list. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

### [filters \(p. 439\)](#)

Provides the specification of a filter used to limit the intents in the response to only those that match the filter specification. You can only specify one filter and only one string to filter on.

Type: Array of [IntentFilter \(p. 823\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

### [maxResults \(p. 439\)](#)

The maximum number of intents to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 439\)](#)

If the response from the `ListIntents` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response.

Use the returned token in the `nextToken` parameter of a `ListIntents` request to return the next page of results. For a complete set of results, call the `ListIntents` operation until the `nextToken` returned in the response is null.

Type: String

Required: No

### [sortBy \(p. 439\)](#)

Determines the sort order for the response from the `ListIntents` operation. You can choose to sort by the intent name or last updated date in either ascending or descending order.

Type: [IntentSortBy \(p. 825\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "intentSummaries": [
        {
            "description": "string",
            "inputContexts": [
                {
                    "name": "string"
                }
            ],
            "intentId": "string",
            "intentName": "string",
            "lastUpdatedDateTime": number,
            "outputContexts": [

```

```
{  
    "name": "string",  
    "timeToLiveInSeconds": number,  
    "turnsToLive": number  
}  
],  
"parentIntentSignature": "string"  
}  
]  
,"localeId": "string",  
"nextToken": "string"  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId** (p. 440)

The identifier of the bot that contains the intent.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion** (p. 440)

The version of the bot that contains the intent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### **intentSummaries** (p. 440)

Summary information for the intents that meet the filter criteria specified in the request. The length of the list is specified in the `maxResults` parameter of the request. If there are more intents available, the `nextToken` field contains a token to get the next page of results.

Type: Array of [IntentSummary](#) (p. 827) objects

### **localeId** (p. 440)

The language and locale of the intents in the list.

Type: String

### **nextToken** (p. 440)

A token that indicates whether there are more results to return in a response to the `ListIntents` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `ListIntents` operation request to get the next page of results.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 936).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListRecommendedIntents

Service: Amazon Lex Model Building V2

Gets a list of recommended intents provided by the bot recommendation that you can use in your bot. Intents in the response are ordered by relevance.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/  
botrecommendations/botRecommendationId/intents HTTP/1.1  
Content-type: application/json  
  
{  
    "maxResults": number,  
    "nextToken": "string"  
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 443)

The unique identifier of the bot associated with the recommended intents.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botRecommendationId** (p. 443)

The identifier of the bot recommendation that contains the recommended intents.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 443)

The version of the bot that contains the recommended intents.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 443)

The identifier of the language and locale of the recommended intents.

Required: Yes

### Request Body

The request accepts the following data in JSON format.

### [maxResults \(p. 443\)](#)

The maximum number of bot recommendations to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 443\)](#)

If the response from the ListRecommendedIntents operation contains more results than specified in the maxResults parameter, a token is returned in the response. Use that token in the nextToken parameter to return the next page of results.

Type: String

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botRecommendationId": "string",
    "botVersion": "string",
    "localeId": "string",
    "nextToken": "string",
    "summaryList": [
        {
            "intentId": "string",
            "intentName": "string",
            "sampleUtterancesCount": number
        }
    ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 444\)](#)

The unique identifier of the bot associated with the recommended intent.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botRecommendationId \(p. 444\)](#)

The identifier of the bot recommendation that contains the recommended intent.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [botVersion \(p. 444\)](#)

The version of the bot that contains the intent.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

#### [localeId \(p. 444\)](#)

The identifier of the language and locale of the intents to list. The string must match one of the supported locales. For more information, see [Supported languages](#).

Type: String

#### [nextToken \(p. 444\)](#)

A token that indicates whether there are more results to return in a response to the ListRecommendedIntents operation. If the nextToken field is present, you send the contents as the nextToken parameter of a ListRecommendedIntents operation request to get the next page of results.

Type: String

#### [summaryList \(p. 444\)](#)

Summary information for the intents that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter of the request. If there are more intents available, the nextToken field contains a token to get the next page of results.

Type: Array of [RecommendedIntentSummary \(p. 848\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### ValidationException

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListSlots

Service: Amazon Lex Model Building V2

Gets a list of slots that match the specified criteria.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/slots/
HTTP/1.1
Content-type: application/json

{
  "filters": [
    {
      "name": "string",
      "operator": "string",
      "values": [ "string" ]
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 447)

The identifier of the bot that contains the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 447)

The version of the bot that contains the slot.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **intentId** (p. 447)

The unique identifier of the intent that contains the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### [localeId \(p. 447\)](#)

The identifier of the language and locale of the slots to list. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### [filters \(p. 447\)](#)

Provides the specification of a filter used to limit the slots in the response to only those that match the filter specification. You can only specify one filter and only one string to filter on.

Type: Array of [SlotFilter \(p. 861\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

### [maxResults \(p. 447\)](#)

The maximum number of slots to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 447\)](#)

If the response from the `ListSlots` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response. Use that token in the `nextToken` parameter to return the next page of results.

Type: String

Required: No

### [sortBy \(p. 447\)](#)

Determines the sort order for the response from the `ListSlots` operation. You can choose to sort by the slot name or last updated date in either ascending or descending order.

Type: [SlotSortBy \(p. 863\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
```

```
"intentId": "string",
"localeId": "string",
"nextToken": "string",
"slotSummaries": [
  {
    "description": "string",
    "lastUpdatedDateTime": number,
    "slotConstraint": "string",
    "slotId": "string",
    "slotName": "string",
    "slotTypeId": "string",
    "valueElicitationPromptSpecification": {
      "allowInterrupt": boolean,
      "maxRetries": number,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              },
              "plainTextMessage": {
                "value": "string"
              },
              "ssmlMessage": {
                "value": "string"
              }
            }
          ]
        }
      ],
      "messageSelectionStrategy": "string",
      "promptAttemptsSpecification": {
```

```
"string" : [
    "allowedInputTypes": {
        "allowAudioInput": boolean,
        "allowDTMFInput": boolean
    },
    "allowInterrupt": boolean,
    "audioAndDTMFInputSpecification": {
        "audioSpecification": {
            "endTimeoutMs": number,
            "maxLengthMs": number
        },
        "dtmfSpecification": {
            "deletionCharacter": "string",
            "endCharacter": "string",
            "endTimeoutMs": number,
            "maxLength": number
        },
        "startTimeoutMs": number
    },
    "textInputSpecification": {
        "startTimeoutMs": number
    }
}
]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 448\)](#)

The identifier of the bot that contains the slots.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 448\)](#)

The version of the bot that contains the slots.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### [intentId \(p. 448\)](#)

The identifier of the intent that contains the slots.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [localeId \(p. 448\)](#)

The language and locale of the slots in the list.

Type: String

### [nextToken \(p. 448\)](#)

A token that indicates whether there are more results to return in a response to the `ListSlots` operation. If the `nextToken` field is present, you send the contents as the `nextToken` parameter of a `ListSlots` operation request to get the next page of results.

Type: String

### [slotSummaries \(p. 448\)](#)

Summary information for the slots that meet the filter criteria specified in the request. The length of the list is specified in the `maxResults` parameter of the request. If there are more slots available, the `nextToken` field contains a token to get the next page of results.

Type: Array of [SlotSummary \(p. 864\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListSlotTypes

Service: Amazon Lex Model Building V2

Gets a list of slot types that match the specified criteria.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/slottypes/ HTTP/1.1
Content-type: application/json

{
  "filters": [
    {
      "name": "string",
      "operator": "string",
      "values": [ "string" ]
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "sortBy": {
    "attribute": "string",
    "order": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 453)

The unique identifier of the bot that contains the slot types.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 453)

The version of the bot that contains the slot type.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId** (p. 453)

The identifier of the language and locale of the slot types to list. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

### [filters \(p. 453\)](#)

Provides the specification of a filter used to limit the slot types in the response to only those that match the filter specification. You can only specify one filter and only one string to filter on.

Type: Array of [SlotTypeFilter \(p. 866\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

### [maxResults \(p. 453\)](#)

The maximum number of slot types to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextToken \(p. 453\)](#)

If the response from the `ListSlotTypes` operation contains more results than specified in the `maxResults` parameter, a token is returned in the response. Use that token in the `nextToken` parameter to return the next page of results.

Type: String

Required: No

### [sortBy \(p. 453\)](#)

Determines the sort order for the response from the `ListSlotTypes` operation. You can choose to sort by the slot type name or last updated date in either ascending or descending order.

Type: [SlotTypeSortBy \(p. 867\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botVersion": "string",
  "localeId": "string",
  "nextToken": "string",
  "slotTypeSummaries": [
    {
      "description": "string",
      "lastUpdatedDateTime": number,
      "parentSlotTypeSignature": "string",
      "slotTypeCategory": "string",
      "slotTypeId": "string",
      "slotTypeName": "string"
    }
  ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId (p. 454)**

The identifier of the bot that contains the slot types.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion (p. 454)**

The version of the bot that contains the slot types.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### **localeId (p. 454)**

The language and local of the slot types in the list.

Type: String

### **nextToken (p. 454)**

A token that indicates whether there are more results to return in a response to the ListSlotTypes operation. If the nextToken field is present, you send the contents as the nextToken parameter of a ListSlotTypes operation request to get the next page of results.

Type: String

### **slotTypeSummaries (p. 454)**

Summary information for the slot types that meet the filter criteria specified in the request. The length of the list is specified in the maxResults parameter of the request. If there are more slot types available, the nextToken field contains a token to get the next page of results.

Type: Array of [SlotTypeSummary \(p. 869\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListTagsForResource

Service: Amazon Lex Model Building V2

Gets a list of tags associated with a resource. Only bots, bot aliases, and bot channels can have tags associated with them.

### Request Syntax

```
GET /tags/resourceARN HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **resourceARN (p. 457)**

The Amazon Resource Name (ARN) of the resource to get a list of tags for.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "tags": [
    {
      "string" : "string"
    }
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

#### **tags (p. 457)**

The tags associated with a resource.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

### Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## SearchAssociatedTranscripts

Service: Amazon Lex Model Building V2

Search for associated transcripts that meet the specified criteria.

### Request Syntax

```
POST /bots/botId/botversions/botVersion/botlocales/localeId/  
botrecommendations/botRecommendationId/associatedtranscripts HTTP/1.1  
Content-type: application/json  
  
{  
    "filters": [  
        {  
            "name": "string",  
            "values": [ "string" ]  
        }  
    ],  
    "maxResults": number,  
    "nextIndex": number,  
    "searchOrder": "string"  
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId (p. 459)**

The unique identifier of the bot associated with the transcripts that you are searching.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botRecommendationId (p. 459)**

The unique identifier of the bot recommendation associated with the transcripts to search.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion (p. 459)**

The version of the bot containing the transcripts that you are searching.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### **localeId (p. 459)**

The identifier of the language and locale of the transcripts to search. The string must match one of the supported locales. For more information, see [Supported languages](#)

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### [filters \(p. 459\)](#)

A list of filter objects.

Type: Array of [AssociatedTranscriptFilter \(p. 739\)](#) objects

Array Members: Fixed number of 1 item.

Required: Yes

### [maxResults \(p. 459\)](#)

The maximum number of bot recommendations to return in each page of results. If there are fewer results than the max page size, only the actual number of results are returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

### [nextIndex \(p. 459\)](#)

If the response from the `SearchAssociatedTranscriptsRequest` operation contains more results than specified in the `maxResults` parameter, an index is returned in the response. Use that index in the `nextIndex` parameter to return the next page of results.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 10000000.

Required: No

### [searchOrder \(p. 459\)](#)

How `SearchResults` are ordered. Valid values are Ascending or Descending. The default is Descending.

Type: String

Valid Values: Ascending | Descending

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "associatedTranscripts": [
        {
            "transcript": "string"
        }
    ],
    "botId": "string",
    "botRecommendationId": "string",
    "botVersion": "string",
```

```
    "localeId": "string",
    "nextIndex": number,
    "totalResults": number
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [associatedTranscripts \(p. 460\)](#)

The object that contains the associated transcript that meet the criteria you specified.

Type: Array of [AssociatedTranscript \(p. 738\)](#) objects

### [botId \(p. 460\)](#)

The unique identifier of the bot associated with the transcripts that you are searching.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botRecommendationId \(p. 460\)](#)

The unique identifier of the bot recommendation associated with the transcripts to search.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 460\)](#)

The version of the bot containing the transcripts that you are searching.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

### [localeId \(p. 460\)](#)

The identifier of the language and locale of the transcripts to search. The string must match one of the supported locales. For more information, see [Supported languages](#)

Type: String

### [nextIndex \(p. 460\)](#)

A index that indicates whether there are more results to return in a response to the SearchAssociatedTranscripts operation. If the nextIndex field is present, you send the contents as the nextIndex parameter of a SearchAssociatedTranscriptsRequest operation to get the next page of results.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 10000000.

### [totalResults \(p. 460\)](#)

The total number of transcripts returned by the search.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## StartBotRecommendation

Service: Amazon Lex Model Building V2

Use this to provide your transcript data, and to start the bot recommendation process.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/botrecommendations/ HTTP/1.1
Content-type: application/json

{
    "encryptionSetting": {
        "associatedTranscriptsPassword": "string",
        "botLocaleExportPassword": "string",
        "kmsKeyArn": "string"
    },
    "transcriptSourceSetting": {
        "s3BucketTranscriptSource": {
            "kmsKeyArn": "string",
            "pathFormat": {
                "objectPrefixes": [ "string" ]
            },
            "s3BucketName": "string",
            "transcriptFilter": {
                "lexTranscriptFilter": {
                    "dateRangeFilter": {
                        "endDateTime": number,
                        "startDateTime": number
                    }
                }
            },
            "transcriptFormat": "string"
        }
    }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 463)

The unique identifier of the bot containing the bot recommendation.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 463)

The version of the bot containing the bot recommendation.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 463)

The identifier of the language and locale of the bot recommendation to start. The string must match one of the supported locales. For more information, see [Supported languages](#)

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### [encryptionSetting \(p. 463\)](#)

The object representing the passwords that will be used to encrypt the data related to the bot recommendation results, as well as the KMS key ARN used to encrypt the associated metadata.

Type: [EncryptionSetting \(p. 796\)](#) object

Required: No

### [transcriptSourceSetting \(p. 463\)](#)

The object representing the Amazon S3 bucket containing the transcript, as well as the associated metadata.

Type: [TranscriptSourceSetting \(p. 888\)](#) object

Required: Yes

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botId": "string",
    "botRecommendationId": "string",
    "botRecommendationStatus": "string",
    "botVersion": "string",
    "creationDateTime": number,
    "encryptionSetting": {
        "associatedTranscriptsPassword": "string",
        "botLocaleExportPassword": "string",
        "kmsKeyArn": "string"
    },
    "localeId": "string",
    "transcriptSourceSetting": {
        "s3BucketTranscriptSource": {
            "kmsKeyArn": "string",
            "pathFormat": {
                "objectPrefixes": [ "string" ]
            },
            "s3BucketName": "string",
            "transcriptFilter": {
                "lexTranscriptFilter": {
                    "dateRangeFilter": {
                        "endDateTime": number,
                        "startDateTime": number
                    }
                }
            },
            "transcriptFormat": "string"
        }
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### **botId (p. 464)**

The unique identifier of the bot containing the bot recommendation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botRecommendationId (p. 464)**

The identifier of the bot recommendation that you have created.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botRecommendationStatus (p. 464)**

The status of the bot recommendation.

If the status is Failed, then the reasons for the failure are listed in the failureReasons field.

Type: String

Valid Values: Processing | Deleting | Deleted | Downloading | Updating | Available | Failed | Stopping | Stopped

### **botVersion (p. 464)**

The version of the bot containing the bot recommendation.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **creationDateTime (p. 464)**

A timestamp of the date and time that the bot recommendation was created.

Type: Timestamp

### **encryptionSetting (p. 464)**

The object representing the passwords that were used to encrypt the data related to the bot recommendation results, as well as the KMS key ARN used to encrypt the associated metadata.

Type: [EncryptionSetting \(p. 796\)](#) object

### **localeId (p. 464)**

The identifier of the language and locale of the bot recommendation to start. The string must match one of the supported locales. For more information, see [Supported languages](#)

Type: String

[transcriptSourceSetting \(p. 464\)](#)

The object representing the Amazon S3 bucket containing the transcript, as well as the associated metadata.

Type: [TranscriptSourceSetting \(p. 888\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## StartImport

Service: Amazon Lex Model Building V2

Starts importing a bot, bot locale, or custom vocabulary from a zip archive that you uploaded to an S3 bucket.

### Request Syntax

```
PUT /imports/ HTTP/1.1
Content-type: application/json

{
  "filePassword": "string",
  "importId": "string",
  "mergeStrategy": "string",
  "resourceSpecification": {
    "botImportSpecification": {
      "botName": "string",
      "botTags": {
        "string" : "string"
      },
      "dataPrivacy": {
        "childDirected": boolean
      },
      "idleSessionTTLInSeconds": number,
      "roleArn": "string",
      "testBotAliasTags": {
        "string" : "string"
      }
    },
    "botLocaleImportSpecification": {
      "botId": "string",
      "botVersion": "string",
      "localeId": "string",
      "nluIntentConfidenceThreshold": number,
      "voiceSettings": {
        "engine": "string",
        "voiceId": "string"
      }
    },
    "customVocabularyImportSpecification": {
      "botId": "string",
      "botVersion": "string",
      "localeId": "string"
    }
  }
}
```

### URI Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### filePassword (p. 468)

The password used to encrypt the zip archive that contains the resource definition. You should always encrypt the zip archive to protect it during transit between your site and Amazon Lex.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

#### [importId \(p. 468\)](#)

The unique identifier for the import. It is included in the response from the [CreateUploadUrl](#) operation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### [mergeStrategy \(p. 468\)](#)

The strategy to use when there is a name conflict between the imported resource and an existing resource. When the merge strategy is FailOnConflict existing resources are not overwritten and the import fails.

Type: String

Valid Values: Overwrite | FailOnConflict | Append

Required: Yes

#### [resourceSpecification \(p. 468\)](#)

Parameters for creating the bot, bot locale or custom vocabulary.

Type: [ImportResourceSpecification \(p. 813\)](#) object

Required: Yes

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "creationDateTime": number,
    "importId": "string",
    "importStatus": "string",
    "mergeStrategy": "string",
    "resourceSpecification": {
        "botImportSpecification": {
            "botName": "string",
            "botTags": {
                "string": "string"
            },
            "dataPrivacy": {
                "childDirected": boolean
            },
            "idleSessionTTLInSeconds": number,
            "roleArn": "string",
            "testBotAliasTags": {
                "string": "string"
            }
        },
        "botLocaleImportSpecification": {
            "botLocale": "string",
            "botName": "string",
            "botTags": {
                "string": "string"
            },
            "dataPrivacy": {
                "childDirected": boolean
            },
            "idleSessionTTLInSeconds": number,
            "roleArn": "string",
            "testBotAliasTags": {
                "string": "string"
            }
        }
    }
}
```

```
        "botId": "string",
        "botVersion": "string",
        "localeId": "string",
        "nluIntentConfidenceThreshold": number,
        "voiceSettings": {
            "engine": "string",
            "voiceId": "string"
        }
    },
    "customVocabularyImportSpecification": {
        "botId": "string",
        "botVersion": "string",
        "localeId": "string"
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### [creationDateTime \(p. 469\)](#)

The date and time that the import request was created.

Type: Timestamp

### [importId \(p. 469\)](#)

A unique identifier for the import.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [importStatus \(p. 469\)](#)

The current status of the import. When the status is Complete the bot, bot alias, or custom vocabulary is ready to use.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

### [mergeStrategy \(p. 469\)](#)

The strategy used when there was a name conflict between the imported resource and an existing resource. When the merge strategy is FailOnConflict existing resources are not overwritten and the import fails.

Type: String

Valid Values: Overwrite | FailOnConflict | Append

### [resourceSpecification \(p. 469\)](#)

The parameters used when importing the resource.

Type: [ImportResourceSpecification \(p. 813\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## StopBotRecommendation

Service: Amazon Lex Model Building V2

Stop an already running Bot Recommendation request.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/  
botrecommendations/botRecommendationId/stopbotrecommendation HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 472)

The unique identifier of the bot containing the bot recommendation to be stopped.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botRecommendationId** (p. 472)

The unique identifier of the bot recommendation to be stopped.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 472)

The version of the bot containing the bot recommendation.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 472)

The identifier of the language and locale of the bot recommendation to stop. The string must match one of the supported locales. For more information, see [Supported languages](#)

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 202  
Content-type: application/json  
{
```

```
"botId": "string",
"botRecommendationId": "string",
"botRecommendationStatus": "string",
"botVersion": "string",
"localeId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### **botId** (p. 472)

The unique identifier of the bot containing the bot recommendation that is being stopped.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botRecommendationId** (p. 472)

The unique identifier of the bot recommendation that is being stopped.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botRecommendationStatus** (p. 472)

The status of the bot recommendation. If the status is Failed, then the reasons for the failure are listed in the failureReasons field.

Type: String

Valid Values: Processing | Deleting | Deleted | Downloading | Updating | Available | Failed | Stopping | Stopped

### **botVersion** (p. 472)

The version of the bot containing the recommendation that is being stopped.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **localeId** (p. 472)

The identifier of the language and locale of the bot response to stop. The string must match one of the supported locales. For more information, see [Supported languages](#)

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## **See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## TagResource

Service: Amazon Lex Model Building V2

Adds the specified tags to the specified resource. If a tag key already exists, the existing value is replaced with the new value.

### Request Syntax

```
POST /tags/resourceARN HTTP/1.1
Content-type: application/json

{
  "tags": {
    "string": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### [resourceARN \(p. 476\)](#)

The Amazon Resource Name (ARN) of the bot, bot alias, or bot channel to tag.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### [tags \(p. 476\)](#)

A list of tag keys to add to the resource. If a tag key already exists, the existing value is replaced with the new value.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Required: Yes

### Response Syntax

```
HTTP/1.1 200
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UntagResource

Service: Amazon Lex Model Building V2

Removes tags from a bot, bot alias, or bot channel.

### Request Syntax

```
DELETE /tags/resourceARN?tagKeys=tagKeys HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### [resourceARN \(p. 478\)](#)

The Amazon Resource Name (ARN) of the resource to remove the tags from.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

#### [tagKeys \(p. 478\)](#)

A list of tag keys to remove from the resource. If a tag key does not exist on the resource, it is ignored.

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

### Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

#### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

#### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateBot

Service: Amazon Lex Model Building V2

Updates the configuration of an existing bot.

### Request Syntax

```
PUT /bots/botId/ HTTP/1.1
Content-type: application/json

{
    "botName": "string",
    "dataPrivacy": {
        "childDirected": boolean
    },
    "description": "string",
    "idleSessionTTLInSeconds": number,
    "roleArn": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 480)

The unique identifier of the bot to update. This identifier is returned by the [CreateBot](#) operation.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **botName** (p. 480)

The new name of the bot. The name must be unique in the account that creates the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### **dataPrivacy** (p. 480)

Provides information on additional privacy protections Amazon Lex should use with the bot's data.

Type: [DataPrivacy](#) (p. 787) object

Required: Yes

#### **description** (p. 480)

A description of the bot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### [idleSessionTTLInSeconds \(p. 480\)](#)

The time, in seconds, that Amazon Lex should keep information about a user's conversation with the bot.

A user interaction remains active for the amount of time specified. If no conversation occurs during this time, the session expires and Amazon Lex deletes any data provided before the timeout.

You can specify between 60 (1 minute) and 86,400 (24 hours) seconds.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 86400.

Required: Yes

#### [roleArn \(p. 480\)](#)

The Amazon Resource Name (ARN) of an IAM role that has permissions to access the bot.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:iam::[0-9]{12}:role/.\*\$

Required: Yes

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "botId": "string",
  "botName": "string",
  "botStatus": "string",
  "creationDateTime": number,
  "dataPrivacy": {
    "childDirected": boolean
  },
  "description": "string",
  "idleSessionTTLInSeconds": number,
  "lastUpdatedDateTime": number,
  "roleArn": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

#### [botId \(p. 481\)](#)

The unique identifier of the bot that was updated.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

#### [botName \(p. 481\)](#)

The name of the bot after the update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

#### [botStatus \(p. 481\)](#)

Shows the current status of the bot. The bot is first in the Creating status. Once the bot is ready for use, it changes to the Available status. After the bot is created, you can use the DRAFT version of the bot.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

#### [creationDateTime \(p. 481\)](#)

A timestamp of the date and time that the bot was created.

Type: Timestamp

#### [dataPrivacy \(p. 481\)](#)

The data privacy settings for the bot after the update.

Type: [DataPrivacy \(p. 787\)](#) object

#### [description \(p. 481\)](#)

The description of the bot after the update.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

#### [idleSessionTTLInSeconds \(p. 481\)](#)

The session timeout, in seconds, for the bot after the update.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 86400.

#### [lastUpdatedDateTime \(p. 481\)](#)

A timestamp of the date and time that the bot was last updated.

Type: Timestamp

#### [roleArn \(p. 481\)](#)

The Amazon Resource Name (ARN) of the IAM role used by the bot after the update.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:iam::[0-9]{12}:role/.\*\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### ConflictException

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### InternalServerException

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### PreconditionFailedException

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### ServiceQuotaExceededException

You have reached a quota for your bot.

HTTP Status Code: 402

### ThrottlingException

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### ValidationException

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateBotAlias

Service: Amazon Lex Model Building V2

Updates the configuration of an existing bot alias.

### Request Syntax

```
PUT /bots/botId/botaliases/botAliasId/ HTTP/1.1
Content-type: application/json

{
    "botAliasLocaleSettings": {
        "string": {
            "codeHookSpecification": {
                "lambdaCodeHook": {
                    "codeHookInterfaceVersion": "string",
                    "lambdaARN": "string"
                }
            },
            "enabled": boolean
        }
    },
    "botAliasName": "string",
    "botVersion": "string",
    "conversationLogSettings": {
        "audioLogSettings": [
            {
                "destination": {
                    "s3Bucket": {
                        "kmsKeyArn": "string",
                        "logPrefix": "string",
                        "s3BucketArn": "string"
                    }
                },
                "enabled": boolean
            }
        ],
        "textLogSettings": [
            {
                "destination": {
                    "cloudWatch": {
                        "cloudWatchLogGroupArn": "string",
                        "logPrefix": "string"
                    }
                },
                "enabled": boolean
            }
        ]
    },
    "description": "string",
    "sentimentAnalysisSettings": {
        "detectSentiment": boolean
    }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botAliasId** (p. 484)

The unique identifier of the bot alias.

Length Constraints: Fixed length of 10.

Pattern: `^(\bTSTALIASID\b|[0-9a-zA-Z]+)$`

Required: Yes

#### **[botId \(p. 484\)](#)**

The identifier of the bot with the updated alias.

Length Constraints: Fixed length of 10.

Pattern: `^[0-9a-zA-Z]+$`

Required: Yes

## **Request Body**

The request accepts the following data in JSON format.

#### **[botAliasLocaleSettings \(p. 484\)](#)**

The new Lambda functions to use in each locale for the bot alias.

Type: String to [BotAliasLocaleSettings \(p. 745\)](#) object map

Map Entries: Maximum number of items.

Required: No

#### **[botAliasName \(p. 484\)](#)**

The new name to assign to the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^([0-9a-zA-Z][_-]?)+$`

Required: Yes

#### **[botVersion \(p. 484\)](#)**

The new bot version to assign to the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: `^(DRAFT|[0-9]+)$`

Required: No

#### **[conversationLogSettings \(p. 484\)](#)**

The new settings for storing conversation logs in Amazon CloudWatch Logs and Amazon S3 buckets.

Type: [ConversationLogSettings \(p. 781\)](#) object

Required: No

#### **[description \(p. 484\)](#)**

The new description to assign to the bot alias.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### [sentimentAnalysisSettings \(p. 484\)](#)

Determines whether Amazon Lex will use Amazon Comprehend to detect the sentiment of user utterances.

Type: [SentimentAnalysisSettings \(p. 856\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botAliasId": "string",
    "botAliasLocaleSettings": {
        "string" : {
            "codeHookSpecification": {
                "lambdaCodeHook": {
                    "codeHookInterfaceVersion": "string",
                    "lambdaARN": "string"
                }
            },
            "enabled": boolean
        }
    },
    "botAliasName": "string",
    "botAliasStatus": "string",
    "botId": "string",
    "botVersion": "string",
    "conversationLogSettings": [
        "audioLogSettings": [
            {
                "destination": {
                    "s3Bucket": {
                        "kmsKeyArn": "string",
                        "logPrefix": "string",
                        "s3BucketArn": "string"
                    }
                },
                "enabled": boolean
            }
        ],
        "textLogSettings": [
            {
                "destination": {
                    "cloudWatch": {
                        "cloudWatchLogGroupArn": "string",
                        "logPrefix": "string"
                    }
                },
                "enabled": boolean
            }
        ]
    },
    "creationDateTime": number,
    "description": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "status": "string",
    "version": "string"
}
```

```
"lastUpdatedDateTime": number,  
"sentimentAnalysisSettings": {  
    "detectSentiment": boolean  
}  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### [botAliasId \(p. 486\)](#)

The identifier of the updated bot alias.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

### [botAliasLocaleSettings \(p. 486\)](#)

The updated Lambda functions to use in each locale for the bot alias.

Type: String to [BotAliasLocaleSettings \(p. 745\)](#) object map

Map Entries: Maximum number of items.

### [botAliasName \(p. 486\)](#)

The updated name of the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

### [botAliasStatus \(p. 486\)](#)

The current status of the bot alias. When the status is Available the alias is ready for use.

Type: String

Valid Values: Creating | Available | Deleting | Failed

### [botId \(p. 486\)](#)

The identifier of the bot with the updated alias.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 486\)](#)

The updated version of the bot that the alias points to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT | [0-9]+)\$

#### [conversationLogSettings \(p. 486\)](#)

The updated settings for storing conversation logs in Amazon CloudWatch Logs and Amazon S3 buckets.

Type: [ConversationLogSettings \(p. 781\)](#) object

#### [creationDateTime \(p. 486\)](#)

A timestamp of the date and time that the bot was created.

Type: Timestamp

#### [description \(p. 486\)](#)

The updated description of the bot alias.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

#### [lastUpdatedDateTime \(p. 486\)](#)

A timestamp of the date and time that the bot was last updated.

Type: Timestamp

#### [sentimentAnalysisSettings \(p. 486\)](#)

Determines whether Amazon Lex will use Amazon Comprehend to detect the sentiment of user utterances.

Type: [SentimentAnalysisSettings \(p. 856\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateBotLocale

Service: Amazon Lex Model Building V2

Updates the settings that a bot has for a specific locale.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/ HTTP/1.1
Content-type: application/json

{
  "description": "string",
  "nluIntentConfidenceThreshold": number,
  "voiceSettings": {
    "engine": "string",
    "voiceId": "string"
  }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 490)

The unique identifier of the bot that contains the locale.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 490)

The version of the bot that contains the locale to be updated. The version can only be the DRAFT version.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 490)

The identifier of the language and locale to update. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **description** (p. 490)

The new description of the locale.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### [nluIntentConfidenceThreshold \(p. 490\)](#)

The new confidence threshold where Amazon Lex inserts the AMAZON.FallbackIntent and AMAZON.KendraSearchIntent intents in the list of possible intents for an utterance.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 1.

Required: Yes

#### [voiceSettings \(p. 490\)](#)

The new Amazon Polly voice Amazon Lex should use for voice interaction with the user.

Type: [VoiceSettings \(p. 890\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
  "botId": "string",
  "botLocaleStatus": "string",
  "botVersion": "string",
  "creationDateTime": number,
  "description": "string",
  "failureReasons": [ "string" ],
  "lastUpdatedDateTime": number,
  "localeId": "string",
  "localeName": "string",
  "nluIntentConfidenceThreshold": number,
  "recommendedActions": [ "string" ],
  "voiceSettings": {
    "engine": "string",
    "voiceId": "string"
  }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

#### [botId \(p. 491\)](#)

The identifier of the bot that contains the updated locale.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

**botLocaleStatus (p. 491)**

The current status of the locale. When the bot status is Built the locale is ready for use.

Type: String

Valid Values: Creating | Building | Built | ReadyExpressTesting | Failed | Deleting | NotBuilt | Importing | Processing

**botVersion (p. 491)**

The version of the bot that contains the updated locale.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

**creationDateTime (p. 491)**

A timestamp of the date and time that the locale was created.

Type: Timestamp

**description (p. 491)**

The updated description of the locale.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

**failureReasons (p. 491)**

If the botLocaleStatus is Failed, the failureReasons field lists the errors that occurred while building the bot.

Type: Array of strings

**lastUpdatedDateTime (p. 491)**

A timestamp of the date and time that the locale was last updated.

Type: Timestamp

**localeId (p. 491)**

The language and locale of the updated bot locale.

Type: String

**localeName (p. 491)**

The updated locale name for the locale.

Type: String

**nluIntentConfidenceThreshold (p. 491)**

The updated confidence threshold for inserting the AMAZON.FallbackIntent and AMAZON.KendraSearchIntent intents in the list of possible intents for an utterance.

Type: Double

Valid Range: Minimum value of 0. Maximum value of 1.

### [recommendedActions \(p. 491\)](#)

Recommended actions to take to resolve an error in the `failureReasons` field.

Type: Array of strings

### [voiceSettings \(p. 491\)](#)

The updated Amazon Polly voice to use for voice interaction with the user.

Type: [VoiceSettings \(p. 890\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateBotRecommendation

Service: Amazon Lex Model Building V2

Updates an existing bot recommendation request.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/  
botrecommendations/botRecommendationId/ HTTP/1.1  
Content-type: application/json  
  
{  
    "encryptionSetting": {  
        "associatedTranscriptsPassword": "string",  
        "botLocaleExportPassword": "string",  
        "kmsKeyArn": "string"  
    }  
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId** (p. 495)

The unique identifier of the bot containing the bot recommendation to be updated.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botRecommendationId** (p. 495)

The unique identifier of the bot recommendation to be updated.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion** (p. 495)

The version of the bot containing the bot recommendation to be updated.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId** (p. 495)

The identifier of the language and locale of the bot recommendation to update. The string must match one of the supported locales. For more information, see [Supported languages](#)

Required: Yes

### Request Body

The request accepts the following data in JSON format.

### encryptionSetting (p. 495)

The object representing the passwords that will be used to encrypt the data related to the bot recommendation results, as well as the KMS key ARN used to encrypt the associated metadata.

Type: [EncryptionSetting \(p. 796\)](#) object

Required: Yes

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botId": "string",
    "botRecommendationId": "string",
    "botRecommendationStatus": "string",
    "botVersion": "string",
    "creationDateTime": number,
    "encryptionSetting": {
        "associatedTranscriptsPassword": "string",
        "botLocaleExportPassword": "string",
        "kmsKeyArn": "string"
    },
    "lastUpdatedDateTime": number,
    "localeId": "string",
    "transcriptSourceSetting": {
        "s3BucketTranscriptSource": {
            "kmsKeyArn": "string",
            "pathFormat": {
                "objectPrefixes": [ "string" ]
            },
            "s3BucketName": "string",
            "transcriptFilter": {
                "lexTranscriptFilter": {
                    "dateRangeFilter": {
                        "endDateTime": number,
                        "startDateTime": number
                    }
                }
            },
            "transcriptFormat": "string"
        }
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### botId (p. 496)

The unique identifier of the bot containing the bot recommendation that has been updated.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botRecommendationId (p. 496)**

The unique identifier of the bot recommendation to be updated.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botRecommendationStatus (p. 496)**

The status of the bot recommendation.

If the status is Failed, then the reasons for the failure are listed in the failureReasons field.

Type: String

Valid Values: Processing | Deleting | Deleted | Downloading | Updating | Available | Failed | Stopping | Stopped

### **botVersion (p. 496)**

The version of the bot containing the bot recommendation that has been updated.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **creationDateTime (p. 496)**

A timestamp of the date and time that the bot recommendation was created.

Type: Timestamp

### **encryptionSetting (p. 496)**

The object representing the passwords that were used to encrypt the data related to the bot recommendation results, as well as the KMS key ARN used to encrypt the associated metadata.

Type: [EncryptionSetting \(p. 796\)](#) object

### **lastUpdatedDateTime (p. 496)**

A timestamp of the date and time that the bot recommendation was last updated.

Type: Timestamp

### **localeId (p. 496)**

The identifier of the language and locale of the bot recommendation to update. The string must match one of the supported locales. For more information, see [Supported languages](#)

Type: String

### **transcriptSourceSetting (p. 496)**

The object representing the Amazon S3 bucket containing the transcript, as well as the associated metadata.

Type: [TranscriptSourceSetting \(p. 888\)](#) object

## **Errors**

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## **See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateExport

Service: Amazon Lex Model Building V2

Updates the password used to protect an export zip archive.

The password is not required. If you don't supply a password, Amazon Lex generates a zip file that is not protected by a password. This is the archive that is available at the pre-signed S3 URL provided by the [DescribeExport](#) operation.

### Request Syntax

```
PUT /exports/exportId/ HTTP/1.1
Content-type: application/json

{
    "filePassword": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **exportId (p. 500)**

The unique identifier Amazon Lex assigned to the export.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **filePassword (p. 500)**

The new password to use to encrypt the export zip archive.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

### Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "creationDateTime": number,
    "exportId": "string",
    "exportStatus": "string",
    "fileFormat": "string",
    "lastUpdatedDateTime": number,
    "resourceSpecification": {
```

```
    "botExportSpecification": {
        "botId": "string",
        "botVersion": "string"
    },
    "botLocaleExportSpecification": {
        "botId": "string",
        "botVersion": "string",
        "localeId": "string"
    },
    "customVocabularyExportSpecification": {
        "botId": "string",
        "botVersion": "string",
        "localeId": "string"
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### [creationDateTime \(p. 500\)](#)

The date and time that the export was created.

Type: Timestamp

### [exportId \(p. 500\)](#)

The unique identifier Amazon Lex assigned to the export.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [exportStatus \(p. 500\)](#)

The status of the export. When the status is Completed the export archive is available for download.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

### [fileFormat \(p. 500\)](#)

The file format used for the files that define the resource. The TSV format is required to export a custom vocabulary only; otherwise use LexJson format.

Type: String

Valid Values: LexJson | TSV

### [lastUpdatedDateTime \(p. 500\)](#)

The date and time that the export was last updated.

Type: Timestamp

### [resourceSpecification \(p. 500\)](#)

A description of the type of resource that was exported, either a bot or a bot locale.

Type: [ExportResourceSpecification \(p. 798\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateIntent

Service: Amazon Lex Model Building V2

Updates the settings for an intent.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/ HTTP/1.1
Content-type: application/json
```

```
{
    "description": "string",
    "dialogCodeHook": {
        "enabled": boolean
    },
    "fulfillmentCodeHook": {
        "active": boolean,
        "enabled": boolean,
        "fulfillmentUpdatesSpecification": {
            "active": boolean,
            "startResponse": {
                "allowInterrupt": boolean,
                "delayInSeconds": number,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            }
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                ],
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    },
                    "plainTextMessage": {
```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"timeoutInSeconds": number,
"updateResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
```

```
"postFulfillmentStatusSpecification": {
    "failureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": string
                },
                "name": string,
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": string,
                        "suppressNextMessage": boolean,
                        "type": string
                    },
                    "intent": {
                        "name": string,
                        "slots": {
                            string : {
                                "shape": string,
                                "value": {
                                    "interpretedValue": string
                                }
                            },
                            "values": [
                                SlotValueOverride
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                string : string
            }
        ],
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": string
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": string,
                                    "value": string
                                }
                            ],
                            "imageUrl": string,
                            "subtitle": string,
                            "title": string
                        },
                        "plainTextMessage": {
                            "value": string
                        },
                        "ssmlMessage": {
                            "value": string
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": string
                            },
                            "imageResponseCard": {
                                "buttons": [

```

```
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  }
},
"response": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        }
      }
    }
  ]
}
```

```
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": "string"
                        }
                    }
                }
            }
        }
    ]
}
}
```

```
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
    ],
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {
                "name": "string",
                "slots": {
                    "string" : {
                        "shape": "string",
                        "value": {
                            "interpretedValue": "string"
                        },
                        "values": [
                            "SlotValueOverride"
                        ]
                    }
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
],
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string": {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string": "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        }
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    }
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
    }
}
```

```
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {

```

```
        "value": "string"
    }
}
]
}
},
"initialResponseSetting": {
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {
                "active": boolean,
                "conditionalBranches": [
                    {
                        "condition": {
                            "expressionString": "string"
                        },
                        "name": "string",
                        "nextStep": {
                            "dialogAction": {
                                "slotToElicit": "string",
                                "suppressNextMessage": boolean,
                                "type": "string"
                            },
                            "intent": {
                                "name": "string",
                                "slots": {
                                    "string" : {
                                        "shape": "string",
                                        "value": {
                                            "interpretedValue": "string"
                                        },
                                        "values": [
                                            "SlotValueOverride"
                                        ]
                                    }
                                }
                            },
                            "sessionAttributes": {
                                "string" : "string"
                            }
                        }
                    }
                ],
                "response": {
                    "allowInterrupt": boolean,
                    "messageGroups": [
                        {
                            "message": {
                                "customPayload": {
                                    "value": "string"
                                },
                                "imageResponseCard": {
                                    "buttons": [
                                        {
                                            "text": "string",
                                            "value": "string"
                                        }
                                    ],
                                    "imageUrl": "string",
                                    "subtitle": "string",
                                    "title": "string"
                                }
                            }
                        }
                    ]
                }
            }
        }
    }
}
```

```
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {

```

```
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                }
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    }
}
```

```
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "successConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
```

```
        "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "title": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        ]
    }
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            }
        }
    ]
}
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        ],
        "defaultBranch": {
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",

```

```
        "suppressNextMessage": boolean,
        "type": string
    },
    "intent": {
        "name": string,
        "slots": {
            string : {
                "shape": string,
                "value": {
                    "interpretedValue": string
                },
                "values": [
                    SlotValueOverride
                ]
            }
        }
    },
    "sessionAttributes": {
        string : string
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": string
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": string,
                            "value": string
                        }
                    ],
                    "imageUrl": string,
                    "subtitle": string,
                    "title": string
                },
                "plainTextMessage": {
                    "value": string
                },
                "ssmlMessage": {
                    "value": string
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": string
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": string,
                                "value": string
                            }
                        ],
                        "imageUrl": string,
                        "subtitle": string,
                        "title": string
                    },
                    "plainTextMessage": {
                        "value": string
                    }
                }
            ]
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
}
},
"conditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {

```

```

        "text": "string",
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {

```

```
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"initialResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
}
```

```
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
```

```
        "string" : "string"
    }
},
"inputContexts": [
{
  "name": "string"
}
],
"intentClosingSetting": {
  "active": boolean,
  "closingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          }
        ]
      }
    ],
    "conditional": {
      "active": boolean,
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    }
  }
}
```

```
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ]
            }
          }
        ]
      }
    ]
  }
]
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
{
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"intentConfirmationSetting": {
    "active": boolean,
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {

```

```
"active": boolean,
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
]
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            },
            "plainTextMessage": {
                "value": "string"
            }
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "text": "string"
            }
        }
    ]
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": [
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            }
                        }
                    }
                }
            }
        }
    ]
}
```

```
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
```

```
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
]
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {

```

```
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    }
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
}
```

```
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    },
    "timeoutNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        }
    }
},
```

```
"intent": {
    "name": "string",
    "slots": {
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
      ],
    }
  },
  "confirmationConditional": {
    "active": boolean,
    "conditionalBranches": [
      {
        "condition": {
          "expressionString": "string"
        },
        "name": "string",
        "nextStep": {
          "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
          },
          "intent": {
            "name": "string",
            "slots": {
              "string" : {
                "shape": "string",
                "value": {
                  "interpretedValue": "string"
                },
                "values": [
                  "SlotValueOverride"
                ]
              }
            }
          },
          "sessionAttributes": {
            "string" : "string"
          }
        }
      },
      "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
          {
            "message": {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              },
              "plainTextMessage": {
                "value": "string"
              },
              "ssmlMessage": {
                "value": "string"
              }
            },
            "variations": [

```

```
        {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        }
      }
    ]
  }
}
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  }
},
"response": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ]
        }
      }
    }
  ]
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"confirmationNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
```

```
"confirmationResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": string
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": string,
                            "value": string
                        }
                    ],
                    "imageUrl": string,
                    "subtitle": string,
                    "title": string
                },
                "plainTextMessage": {
                    "value": string
                },
                "ssmlMessage": {
                    "value": string
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": string
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": string,
                                "value": string
                            }
                        ],
                        "imageUrl": string,
                        "subtitle": string,
                        "title": string
                    },
                    "plainTextMessage": {
                        "value": string
                    },
                    "ssmlMessage": {
                        "value": string
                    }
                }
            ]
        }
    ],
    "declinationConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": string
                },
                "name": string,
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": string,
                        "suppressNextMessage": boolean,
                        "type": string
                    }
                }
            }
        ]
    }
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    }
}
```

```
        }
      ]
    }
  ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [

```

```
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
},
"declinationNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"declinationResponse": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        }
      },
      "plainTextMessage": {

```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
{
    "condition": {
        "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string": {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    }
},
"sessionAttributes": {
```

```

        "string" : "string"
    },
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "suppressNextMessage": boolean,
                "type": "string"
            },
            "intent": {

```

```
"name": "string",
"slots": {
    "string" : {
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ]
      }
    ]
  }
},
"failureNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"failureResponse": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      },
      "variations": [
        {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"promptSpecification": {
    "allowInterrupt": boolean,
    "maxRetries": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
      ],
    },
  ],
  "messageSelectionStrategy": "string",
  "promptAttemptsSpecification": {
    "string" : {
      "allowedInputTypes": {
        "allowAudioInput": boolean,
        "allowDTMFInput": boolean
      },
      "allowInterrupt": boolean,
      "audioAndDTMFInputSpecification": {
        "audioSpecification": {
          "endTimeTimeoutMs": number,
          "maxLengthMs": number
        },
        "dtmfSpecification": {
          "deletionCharacter": "string",
          "endCharacter": "string",
          "endTimeTimeoutMs": number,
          "maxLength": number
        },
        "startTimeTimeoutMs": number
      },
      "textInputSpecification": {
        "startTimeTimeoutMs": number
      }
    }
  }
},
"intentName": "string",
"kendraConfiguration": {
  "kendraIndex": "string",
  "queryFilterString": "string",
  "queryFilterStringEnabled": boolean
},
"outputContexts": [
  {
    "name": "string",
    "timeToLiveInSeconds": number,
    "turnsToLive": number
  }
],
"parentIntentSignature": "string",
"sampleUtterances": [
  {
    "utterance": "string"
  }
],
"slotPriorities": [
  {
    "priority": number,
    "slotId": "string"
  }
]
}
```

## URI Request Parameters

The request uses the following URI parameters.

### **botId (p. 503)**

The identifier of the bot that contains the intent.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **botVersion (p. 503)**

The version of the bot that contains the intent. Must be DRAFT.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

### **intentId (p. 503)**

The unique identifier of the intent to update.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **localeId (p. 503)**

The identifier of the language and locale where this intent is used. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

## **Request Body**

The request accepts the following data in JSON format.

### **description (p. 503)**

The new description of the intent.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

### **dialogCodeHook (p. 503)**

The new Lambda function to use between each turn of the conversation with the bot.

Type: [DialogCodeHookSettings \(p. 792\)](#) object

Required: No

### **fulfillmentCodeHook (p. 503)**

The new Lambda function to call when all of the intents required slots are provided and the intent is ready for fulfillment.

Type: [FulfillmentCodeHookSettings \(p. 804\)](#) object

Required: No

#### [initialResponseSetting \(p. 503\)](#)

Type: [InitialResponseSetting \(p. 817\)](#) object

Required: No

#### [inputContexts \(p. 503\)](#)

A new list of contexts that must be active in order for Amazon Lex to consider the intent.

Type: Array of [InputContext \(p. 818\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

Required: No

#### [intentClosingSetting \(p. 503\)](#)

The new response that Amazon Lex sends the user when the intent is closed.

Type: [IntentClosingSetting \(p. 819\)](#) object

Required: No

#### [intentConfirmationSetting \(p. 503\)](#)

New prompts that Amazon Lex sends to the user to confirm the completion of an intent.

Type: [IntentConfirmationSetting \(p. 820\)](#) object

Required: No

#### [intentName \(p. 503\)](#)

The new name for the intent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### [kendraConfiguration \(p. 503\)](#)

New configuration settings for connecting to an Amazon Kendra index.

Type: [KendraConfiguration \(p. 829\)](#) object

Required: No

#### [outputContexts \(p. 503\)](#)

A new list of contexts that Amazon Lex activates when the intent is fulfilled.

Type: Array of [OutputContext \(p. 837\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

#### [parentIntentSignature \(p. 503\)](#)

The signature of the new built-in intent to use as the parent of this intent.

Type: String

Required: No

#### [sampleUtterances \(p. 503\)](#)

New utterances used to invoke the intent.

Type: Array of [SampleUtterance \(p. 854\)](#) objects

Required: No

#### [slotPriorities \(p. 503\)](#)

A new list of slots and their priorities that are contained by the intent.

Type: Array of [SlotPriority \(p. 862\)](#) objects

Required: No

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "botId": "string",
  "botVersion": "string",
  "creationDateTime": number,
  "description": "string",
  "dialogCodeHook": {
    "enabled": boolean
  },
  "fulfillmentCodeHook": {
    "active": boolean,
    "enabled": boolean,
    "fulfillmentUpdatesSpecification": {
      "active": boolean,
      "startResponse": {
        "allowInterrupt": boolean,
        "delayInSeconds": number,
        "messageGroups": [
          {
            "message": {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              },
              "plainTextMessage": {
                "value": "string"
              },
              "ssmlMessage": {
                "value": "string"
              }
            }
          }
        ]
      }
    }
  }
}
```

```
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        ]
    },
    "timeoutInSeconds": number,
    "updateResponse": {
        "allowInterrupt": boolean,
        "frequencyInSeconds": number,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ]
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
},
"postFulfillmentStatusSpecification": {
    "failureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                        "type": "string"
                    },
                    "intent": {
                        "name": "string",
                        "slots": {
                            "string" : {
                                "shape": "string",
                                "value": {
                                    "interpretedValue": "string"
                                },
                                "values": [
                                    "SlotValueOverride"
                                ]
                            }
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "image": "string"
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

```
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
```

```
"allowInterrupt": boolean,
"messageGroups": [
  {
    "message": {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  }
},
"failureNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "string": "string"
        }
      }
    }
  }
}
```

```
        "interpretedValue": "string"
    },
    "values": [
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "successConditional": {
```

```
"active": boolean,
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
]
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
"imageResponseCard": {
    "buttons": [
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"timeoutConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            }
                        }
                    }
                }
            }
        }
    ]
}
```

```
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
```

```
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
}
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
            ]
        }
    ]
}
```

```
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"initialResponseSetting": {
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {
                "active": boolean,
                "conditionalBranches": [
                    {
                        "condition": {
                            "expressionString": "string"
                        },
                        "name": "string",
                        "nextStep": {
                            "dialogAction": {
                                "slotToElicit": "string",
                                "suppressNextMessage": boolean,
                                "type": "string"
                            },
                            "intent": {
                                "name": "string",
                                "slots": {
                                    "string" : {
                                        "shape": "string",
                                        "value": {
                                            "interpretedValue": "string"
                                        },
                                        "values": [
                                            "SlotValueOverride"
                                        ]
                                    }
                                }
                            },
                            "sessionAttributes": {
                                "string" : "string"
                            }
                        }
                    }
                ]
            }
        }
    }
},
"response": {
```

```
"allowInterrupt": boolean,
"messageGroups": [
  {
    "message": {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  }
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",

```

```
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
    },
    "failureNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "failureResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": 'string'
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                }
            ]
        }
    }
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
},
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "imageUrl": "string",
                                "subtitle": "string",
                                "title": "string"
                            },
                            "plainTextMessage": {
                                "value": "string"
                            }
                        }
                    }
                ]
            }
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string": {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string": "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    }
                }
            }
        ]
    }
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string": {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "timeoutConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "branch": {
                    "actions": [
                        ...
                    ],
                    "branchId": "string"
                }
            }
        ]
    }
}
```

```
"name": "string",
"nextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"response": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      },
      "variations": [
        {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          }
        }
      ]
    }
  ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
}
},
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [

```

```
{  
    "customPayload": {  
        "value": "string"  
    },  
    "imageResponseCard": {  
        "buttons": [  
            {  
                "text": "string",  
                "value": "string"  
            }  
        ],  
        "imageUrl": "string",  
        "subtitle": "string",  
        "title": "string"  
    },  
    "plainTextMessage": {  
        "value": "string"  
    },  
    "ssmlMessage": {  
        "value": "string"  
    }  
}  
]  
}  
}  
}  
},  
"timeoutNextStep": {  
    "dialogAction": {  
        "slotToElicit": "string",  
        "suppressNextMessage": boolean,  
        "type": "string"  
    },  
    "intent": {  
        "name": "string",  
        "slots": {  
            "string" : {  
                "shape": "string",  
                "value": {  
                    "interpretedValue": "string"  
                },  
                "values": [  
                    "SlotValueOverride"  
                ]  
            }  
        }  
    },  
    "sessionAttributes": {  
        "string" : "string"  
    }  
},  
"timeoutResponse": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"conditional": {
    "active": boolean,
    "conditionalBranches": [
{
    "condition": {
        "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    }
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    }
}
]
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "type": "string"
            }
        }
    }
}
```

```
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    ]
}
]
},
"initialResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",

```

```
        "suppressNextMessage": boolean,
        "type": string
    },
    "intent": {
        "name": string,
        "slots": {
            string : {
                "shape": string,
                "value": {
                    "interpretedValue": string
                },
                "values": [
                    SlotValueOverride
                ]
            }
        }
    },
    "sessionAttributes": {
        string : string
    }
}
],
"inputContexts": [
{
    "name": string
}
],
"intentClosingSetting": {
    "active": boolean,
    "closingResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": string
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": string,
                                "value": string
                            }
                        ],
                        "imageUrl": string,
                        "subtitle": string,
                        "title": string
                    }
                },
                "plainTextMessage": {
                    "value": string
                },
                "ssmlMessage": {
                    "value": string
                }
            }
        ],
        "variations": [
            {
                "customPayload": {
                    "value": string
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": string,
                            "value": string
                        }
                    ]
                }
            }
        ]
    }
}
```

```
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"conditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                },
                "sessionAttributes": {
                    "string" : "string"
                }
            },
            "response": {
                "allowInterrupt": boolean,
                "messageGroups": [
                    {
                        "message": {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "imageUrl": "string",
                                "subtitle": "string",
                                "title": "string"
                            }
                        }
                    }
                ]
            }
        }
    ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
        {

```

```
"message": {
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"nextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [

```

```
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
}
],
"intentConfirmationSetting": {
    "active": boolean,
    "codeHook": {
        "active": boolean,
        "enableCodeHookInvocation": boolean,
        "invocationLabel": "string",
        "postCodeHookSpecification": {
            "failureConditional": {
                "active": boolean,
                "conditionalBranches": [
                    {
                        "condition": {
                            "expressionString": "string"
                        },
                        "name": "string",
                        "nextStep": {
                            "dialogAction": {
                                "slotToElicit": "string",
                                "suppressNextMessage": boolean,
                                "type": "string"
                            },
                            "intent": {
                                "name": "string",
                                "slots": {
                                    "string" : {
                                        "shape": "string",
                                        "value": {
                                            "interpretedValue": "string"
                                        },
                                        "values": [
                                            "SlotValueOverride"
                                        ]
                                    }
                                }
                            }
                        }
                    }
                ],
                "sessionAttributes": {
                    "string" : "string"
                }
            }
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    }
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [

```

```
{  
    "message": {  
        "customPayload": {  
            "value": "string"  
        },  
        "imageResponseCard": {  
            "buttons": [  
                {  
                    "text": "string",  
                    "value": "string"  
                }  
            ],  
            "imageUrl": "string",  
            "subtitle": "string",  
            "title": "string"  
        },  
        "plainTextMessage": {  
            "value": "string"  
        },  
        "ssmlMessage": {  
            "value": "string"  
        }  
    },  
    "variations": [  
        {  
            "customPayload": {  
                "value": "string"  
            },  
            "imageResponseCard": {  
                "buttons": [  
                    {  
                        "text": "string",  
                        "value": "string"  
                    }  
                ],  
                "imageUrl": "string",  
                "subtitle": "string",  
                "title": "string"  
            },  
            "plainTextMessage": {  
                "value": "string"  
            },  
            "ssmlMessage": {  
                "value": "string"  
            }  
        }  
    ]  
},  
"failureNextStep": {  
    "dialogAction": {  
        "slotToElicit": "string",  
        "suppressNextMessage": boolean,  
        "type": "string"  
    },  
    "intent": {  
        "name": "string",  
        "slots": {  
            "string" : {  
                "shape": "string",  
                "value": {  
                    "interpretedValue": "string"  
                },  
            }  
        }  
    }  
},  
"intents": [  
    {  
        "name": "string",  
        "slots": {  
            "string" : {  
                "shape": "string",  
                "value": {  
                    "interpretedValue": "string"  
                },  
            }  
        }  
    }  
],  
"slotTypes": [  
    {  
        "name": "string",  
        "values": [  
            {  
                "alias": "string",  
                "value": "string"  
            }  
        ]  
    }  
],  
"types": [  
    {  
        "name": "string",  
        "values": [  
            {  
                "alias": "string",  
                "value": "string"  
            }  
        ]  
    }  
]
```

```
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "successConditional": {
        "active": boolean,
        "conditionalBranches": [

```

```
{  
    "condition": {  
        "expressionString": "string"  
    },  
    "name": "string",  
    "nextStep": {  
        "dialogAction": {  
            "slotToElicit": "string",  
            "suppressNextMessage": boolean,  
            "type": "string"  
        },  
        "intent": {  
            "name": "string",  
            "slots": {  
                "string" : {  
                    "shape": "string",  
                    "value": {  
                        "interpretedValue": "string"  
                    },  
                    "values": [  
                        "SlotValueOverride"  
                    ]  
                }  
            }  
        },  
        "sessionAttributes": {  
            "string" : "string"  
        }  
    },  
    "response": {  
        "allowInterrupt": boolean,  
        "messageGroups": [  
            {  
                "message": {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    },  
                    "ssmlMessage": {  
                        "value": "string"  
                    }  
                },  
                "variations": [  
                    {  
                        "customPayload": {  
                            "value": "string"  
                        },  
                        "imageResponseCard": {  
                            "buttons": [  
                                {  
                                    "text": "string",  
                                    "value": "string"  
                                }  
                            ]  
                        }  
                    }  
                ]  
            }  
        ]  
    }  
}
```

```
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {

```

```
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
}
},
"successNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [

```

```
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  },
  "variations": [
    {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    }
  ]
},
"timeoutConditional": {
  "active": boolean,
  "conditionalBranches": [
    {
      "condition": {
        "expressionString": "string"
      },
      "name": "string",
      "nextStep": {
        "dialogAction": {
          "slotToElicit": "string",
          "suppressNextMessage": boolean,
          "type": "string"
        },
        "intent": {
          "name": "string",
          "slots": {
            "string" : {
              "shape": "string",
              "value": {
                "interpretedValue": "string"
              }
            },
            "values": [

```

```
        "SlotValueOverride"
    ]
}
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"defaultBranch": {
    "nextStep": {
```

```
"dialogAction": {  
    "slotToElicit": "string",  
    "suppressNextMessage": boolean,  
    "type": "string"  
},  
"intent": {  
    "name": "string",  
    "slots": {  
        "string" : {  
            "shape": "string",  
            "value": {  
                "interpretedValue": "string"  
            },  
            "values": [  
                "SlotValueOverride"  
            ]  
        }  
    }  
},  
"sessionAttributes": {  
    "string" : "string"  
}  
},  
"response": {  
    "allowInterrupt": boolean,  
    "messageGroups": [  
        {  
            "message": {  
                "customPayload": {  
                    "value": "string"  
                },  
                "imageResponseCard": {  
                    "buttons": [  
                        {  
                            "text": "string",  
                            "value": "string"  
                        }  
                    ],  
                    "imageUrl": "string",  
                    "subtitle": "string",  
                    "title": "string"  
                },  
                "plainTextMessage": {  
                    "value": "string"  
                },  
                "ssmlMessage": {  
                    "value": "string"  
                }  
            },  
            "variations": [  
                {  
                    "customPayload": {  
                        "value": "string"  
                    },  
                    "imageResponseCard": {  
                        "buttons": [  
                            {  
                                "text": "string",  
                                "value": "string"  
                            }  
                        ],  
                        "imageUrl": "string",  
                        "subtitle": "string",  
                        "title": "string"  
                    },  
                    "plainTextMessage": {  
                        "value": "string"  
                    }  
                }  
            ]  
        }  
    ]  
},  
"slotType": {  
    "description": "string",  
    "name": "string",  
    "parentType": "string",  
    "patterns": [  
        "string"  
    ],  
    "values": [  
        "string"  
    ]  
},  
"slotValue": {  
    "interpretedValue": "string",  
    "originalValue": "string",  
    "value": "string"  
},  
"slotValueOverrides": {  
    "string" : "string"  
},  
"variables": {  
    "string" : "string"  
},  
"version": "string"
```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
}
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
}
},
"confirmationConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string" : "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        }
    }
}
```

```
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    }
                ]
            }
        }
    },
    "confirmationNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        }
    }
}
```

```
"intent": {
    "name": "string",
    "slots": {
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"confirmationResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        ]
      ],
    ],
  },
  "declinationConditional": {
    "active": boolean,
    "conditionalBranches": [
      {
        "condition": {
          "expressionString": "string"
        },
        "name": "string",
        "nextStep": {
          "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
          },
          "intent": {
            "name": "string",
            "slots": {
              "string" : {
                "shape": "string",
                "value": {
                  "interpretedValue": "string"
                },
                "values": [
                  "SlotValueOverride"
                ]
              }
            }
          }
        },
        "sessionAttributes": {
          "string" : "string"
        }
      },
      "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
          {
            "message": {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
              },
              "plainTextMessage": {
                "value": "string"
              },
              "ssmlMessage": {
                "value": "string"
              }
            },
            "variations": [
              {
                "customPayload": {

```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                }
            }
        ]
    }
}
```

```
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"declinationNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"declinationResponse": {
    "allowInterrupt": boolean,

```

```
"messageGroups": [
    {
        "message": {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        },
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            }
        ]
    }
},
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "type": "string"
                }
            }
        }
    ]
}
```

```
        "suppressNextMessage": boolean,
        "type": string
    },
    "intent": {
        "name": string,
        "slots": {
            string : {
                "shape": string,
                "value": {
                    "interpretedValue": string
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        string : string
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": string
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": string,
                            "value": string
                        }
                    ],
                    "imageUrl": string,
                    "subtitle": string,
                    "title": string
                },
                "plainTextMessage": {
                    "value": string
                },
                "ssmlMessage": {
                    "value": string
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": string
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": string,
                                "value": string
                            }
                        ],
                        "imageUrl": string,
                        "subtitle": string,
                        "title": string
                    },
                    "plainTextMessage": {
                        "value": string
                    }
                },
                {
                    "customPayload": {
                        "value": string
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": string,
                                "value": string
                            }
                        ],
                        "imageUrl": string,
                        "subtitle": string,
                        "title": string
                    },
                    "plainTextMessage": {
                        "value": string
                    }
                }
            ]
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },

```

```
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"promptSpecification": {
    "allowInterrupt": boolean,
    "maxRetries": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
],
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"intenId": "string",
"intenName": "string",
"kendraConfiguration": {
    "kendraIndex": "string",
    "queryFilterString": "string",
    "queryFilterStringEnabled": boolean
},
"lastUpdatedDateTime": number,
"localeId": "string",
"outputContexts": [
    {
        "name": "string",
        "timeToLiveInSeconds": number,
        "turnsToLive": number
    }
],

```

```
"parentIntentSignature": "string",
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"slotPriorities": [
    {
        "priority": number,
        "slotId": "string"
    }
]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botId (p. 561)**

The identifier of the bot that contains the intent.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **botVersion (p. 561)**

The version of the bot that contains the intent. Will always be DRAFT.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### **creationDateTime (p. 561)**

A timestamp of when the intent was created.

Type: Timestamp

### **description (p. 561)**

The updated description of the intent.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

### **dialogCodeHook (p. 561)**

The updated Lambda function called during each turn of the conversation with the user.

Type: [DialogCodeHookSettings \(p. 792\)](#) object

### **fulfillmentCodeHook (p. 561)**

The updated Lambda function called when the intent is ready for fulfillment.

Type: [FulfillmentCodeHookSettings \(p. 804\)](#) object

[initialResponseSetting \(p. 561\)](#)

Type: [InitialResponseSetting \(p. 817\)](#) object

[inputContexts \(p. 561\)](#)

The updated list of contexts that must be active for the intent to be considered by Amazon Lex.

Type: Array of [InputContext \(p. 818\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

[intentClosingSetting \(p. 561\)](#)

The updated response that Amazon Lex sends the user when the intent is closed.

Type: [IntentClosingSetting \(p. 819\)](#) object

[intentConfirmationSetting \(p. 561\)](#)

The updated prompts that Amazon Lex sends to the user to confirm the completion of an intent.

Type: [IntentConfirmationSetting \(p. 820\)](#) object

[intentId \(p. 561\)](#)

The identifier of the intent that was updated.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

[intentName \(p. 561\)](#)

The updated name of the intent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

[kendraConfiguration \(p. 561\)](#)

The updated configuration for connecting to an Amazon Kendra index with the AMAZON.KendraSearchIntent intent.

Type: [KendraConfiguration \(p. 829\)](#) object

[lastUpdatedDateTime \(p. 561\)](#)

A timestamp of the last time that the intent was modified.

Type: Timestamp

[localeId \(p. 561\)](#)

The updated language and locale of the intent.

Type: String

[outputContexts \(p. 561\)](#)

The updated list of contexts that Amazon Lex activates when the intent is fulfilled.

Type: Array of [OutputContext \(p. 837\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

#### **parentIntentSignature (p. 561)**

The updated built-in intent that is the parent of this intent.

Type: String

#### **sampleUtterances (p. 561)**

The updated list of sample utterances for the intent.

Type: Array of [SampleUtterance \(p. 854\)](#) objects

#### **slotPriorities (p. 561)**

The updated list of slots and their priorities that are elicited from the user for the intent.

Type: Array of [SlotPriority \(p. 862\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateResourcePolicy

Service: Amazon Lex Model Building V2

Replaces the existing resource policy for a bot or bot alias with a new one. If the policy doesn't exist, Amazon Lex returns an exception.

### Request Syntax

```
PUT /policy/resourceArn?expectedRevisionId=expectedRevisionId HTTP/1.1
Content-type: application/json

{
  "policy": "string"
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **expectedRevisionId** (p. 621)

The identifier of the revision of the policy to update. If this revision ID doesn't match the current revision ID, Amazon Lex throws an exception.

If you don't specify a revision, Amazon Lex overwrites the contents of the policy with the new values.

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

#### **resourceArn** (p. 621)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

### Request Body

The request accepts the following data in JSON format.

#### **policy** (p. 621)

A resource policy to add to the resource. The policy is a JSON structure that contains one or more statements that define the policy. The policy must follow the IAM syntax. For more information about the contents of a JSON policy document, see [IAM JSON policy reference](#).

If the policy isn't valid, Amazon Lex returns a validation exception.

Type: String

Length Constraints: Minimum length of 2.

Required: Yes

### Response Syntax

```
HTTP/1.1 200
```

```
Content-type: application/json

{
    "resourceArn": "string",
    "revisionId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [resourceArn \(p. 621\)](#)

The Amazon Resource Name (ARN) of the bot or bot alias that the resource policy is attached to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

### [revisionId \(p. 621\)](#)

The current revision of the resource policy. Use the revision ID to make sure that you are updating the most current version of a resource policy when you add a policy statement to a resource, delete a resource, or update a resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **InternalServerError**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ResourceNotFoundException**

You asked to describe a resource that doesn't exist. Check the resource that you are requesting and try again.

HTTP Status Code: 404

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateSlot

Service: Amazon Lex Model Building V2

Updates the settings for a slot.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/intents/intentId/slots/slotId
HTTP/1.1
Content-type: application/json

{
  "description": "string",
  "multipleValuesSetting": {
    "allowMultipleValues": boolean
  },
  "obfuscationSetting": {
    "obfuscationSettingType": "string"
  },
  "slotName": "string",
  "slotTypeId": "string",
  "subSlotSetting": {
    "expression": "string",
    "slotSpecifications": {
      "string" : {
        "slotTypeId": "string",
        "valueElicitationSetting": {
          "defaultValueSpecification": {
            "defaultValueList": [
              {
                "defaultValue": "string"
              }
            ]
          },
          "promptSpecification": {
            "allowInterrupt": boolean,
            "maxRetries": number,
            "messageGroups": [
              {
                "message": {
                  "customPayload": {
                    "value": "string"
                  },
                  "imageResponseCard": {
                    "buttons": [
                      {
                        "text": "string",
                        "value": "string"
                      }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                  },
                  "plainTextMessage": {
                    "value": "string"
                  },
                  "ssmlMessage": {
                    "value": "string"
                  }
                },
                "variations": [
                  {
                    "customPayload": {

```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {

```

```
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  },
  "stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          }
        }
      }
    ]
  }
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [

```

```
        {
            "text": "string",
            "value": "string"
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
}
}
},
"valueElicitationSetting": {
    "defaultValueSpecification": {
        "defaultValueList": [
            {
                "defaultValue": "string"
            }
        ]
    },
    "promptSpecification": {
        "allowInterrupt": boolean,
        "maxRetries": number,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
],
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeTimeoutMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeTimeoutMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"slotCaptureSetting": {
    "captureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
                        "slotToElicit": "string",
                        "suppressNextMessage": boolean,
                        "type": "string"
                    }
                }
            }
        ]
    }
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
}
```

```
        }
      ]
    }
  ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [

```

```
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
},
"captureNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"captureResponse": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {

```

```

        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
"codeHook": {
    "active": boolean,
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string",
    "postCodeHookSpecification": {
        "failureConditional": {
            "active": boolean,
            "conditionalBranches": [
                {
                    "condition": {
                        "expressionString": "string"
                    },
                    "name": "string",
                    "nextStep": {
                        "dialogAction": {
                            "slotToElicit": "string",
                            "suppressNextMessage": boolean,
                            "type": "string"
                        },
                        "intent": {
                            "name": "string",
                            "slots": {
                                "string" : {
                                    "shape": "string",
                                    "value": {
                                        "interpretedValue": "string"
                                    }
                                },
                                "values": [
                                    "SlotValueOverride"
                                ]
                            }
                        }
                    }
                }
            ]
        }
    }
}
]
```

```
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
},
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        }
    }
},
```

```
"intent": {
    "name": "string",
    "slots": {
        "string" : {
            "shape": "string",
            "value": {
                "interpretedValue": "string"
            },
            "values": [
                "SlotValueOverride"
            ]
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
      ]
    }
  ],
  "failureNextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": [
            "interpretedValue": "string"
          ],
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            }
          }
        ]
      }
    ]
  }
}
```

```
        "value": "string"
    }
],
"imageUrl": "string",
"subtitle": "string",
"title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
"successConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string": {
                            "shape": "string",
                            "value": {
                                "interpretedValue": "string"
                            },
                            "values": [
                                "SlotValueOverride"
                            ]
                        }
                    }
                }
            },
            "sessionAttributes": {
                "string": "string"
            }
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "type": "string"
                        }
                    }
                }
            ]
        }
    ]
}
```

```
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "maxAttempts": integer,
        "reprompt": {
            "maxAttempts": integer,
            "text": "string"
        }
    }
}
```

```
"messageGroups": [
  {
    "message": {
      "customPayload": {
        "value": "string"
      },
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    },
    "variations": [
      {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      }
    ]
  }
},
"successNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string": {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        }
      }
    }
  }
}
```

```
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "timeoutConditional": {
        "active": boolean,
        "value": "string"
    }
}
```

```
"conditionalBranches": [
  {
    "condition": {
      "expressionString": "string"
    },
    "name": "string",
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          },
          "plainTextMessage": {
            "value": "string"
          },
          "ssmlMessage": {
            "value": "string"
          }
        },
        "variations": [
          {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ]
            }
          }
        ]
      }
    ]
  }
]
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
],
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                }
            }
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
{
    "timeoutNextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "timeoutResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {

```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",

```

```
"slots": {
    "string" : {
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        }
      ],
    },
  ],
  "defaultBranch": {
    "nextStep": {
      "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
      },
      "intent": {
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "value": {
              "interpretedValue": "string"
            },
            "values": [
              "SlotValueOverride"
            ]
          }
        }
      },
      "sessionAttributes": {
        "string" : "string"
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ],
              "imageUrl": "string",
              "subtitle": "string",
              "title": "string"
            },
            "plainTextMessage": {
              "value": "string"
            },
            "ssmlMessage": {
              "value": "string"
            }
          },
          "variations": [
            {
              "customPayload": {
                "value": "string"
              },
              "imageResponseCard": {
                "buttons": [
                  {
                    "text": "string",
                    "value": "string"
                  }
                ]
              }
            }
          ]
        }
      ]
    }
  }
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
}
]
}
}
},
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {

```

```
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"slotConstraint": "string",
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            }
        ],
        "variations": [
            {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
```

```
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
]
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
],
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
}
```

}

## URI Request Parameters

The request uses the following URI parameters.

### **botId (p. 624)**

The unique identifier of the bot that contains the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **botVersion (p. 624)**

The version of the bot that contains the slot. Must always be DRAFT.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

### **intentId (p. 624)**

The identifier of the intent that contains the slot.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **localeId (p. 624)**

The identifier of the language and locale that contains the slot. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

### **slotId (p. 624)**

The unique identifier for the slot to update.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### **description (p. 624)**

The new description for the slot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### [multipleValuesSetting \(p. 624\)](#)

Determines whether the slot accepts multiple values in one response. Multiple value slots are only available in the en-US locale. If you set this value to true in any other locale, Amazon Lex throws a ValidationException.

If the multipleValuesSetting is not set, the default value is false.

Type: [MultipleValuesSetting \(p. 834\)](#) object

Required: No

#### [obfuscationSetting \(p. 624\)](#)

New settings that determine how slot values are formatted in Amazon CloudWatch logs.

Type: [ObfuscationSetting \(p. 836\)](#) object

Required: No

#### [slotName \(p. 624\)](#)

The new name for the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### [slotTypeId \(p. 624\)](#)

The unique identifier of the new slot type to associate with this slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern: ^((AMAZON\.)[a-zA-Z\_]+?|[0-9a-zA-Z]+)\$

Required: No

#### [subSlotSetting \(p. 624\)](#)

Specifications for the constituent sub slots and the expression for the composite slot.

Type: [SubSlotSetting \(p. 881\)](#) object

Required: No

#### [valueElicitationSetting \(p. 624\)](#)

A new set of prompts that Amazon Lex sends to the user to elicit a response the provides a value for the slot.

Type: [SlotValueElicitationSetting \(p. 873\)](#) object

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "creationDateTime": number,
    "description": "string",
    "intentId": "string",
    "lastUpdatedDateTime": number,
    "localeId": "string",
    "multipleValuesSetting": {
        "allowMultipleValues": boolean
    },
    "obfuscationSetting": {
        "obfuscationSettingType": "string"
    },
    "slotId": "string",
    "slotName": "string",
    "slotTypeId": "string",
    "subSlotSetting": {
        "expression": "string",
        "slotSpecifications": {
            "string" : {
                "slotTypeId": "string",
                "valueElicitationSetting": {
                    "defaultValueSpecification": {
                        "defaultValueList": [
                            {
                                "defaultValue": "string"
                            }
                        ]
                    },
                    "promptSpecification": {
                        "allowInterrupt": boolean,
                        "maxRetries": number,
                        "messageGroups": [
                            {
                                "message": {
                                    "customPayload": {
                                        "value": "string"
                                    },
                                    "imageResponseCard": {
                                        "buttons": [
                                            {
                                                "text": "string",
                                                "value": "string"
                                            }
                                        ],
                                        "imageUrl": "string",
                                        "subtitle": "string",
                                        "title": "string"
                                    },
                                    "plainTextMessage": {
                                        "value": "string"
                                    },
                                    "ssmlMessage": {
                                        "value": "string"
                                    }
                                }
                            ],
                            "slot": "string"
                        ]
                    }
                }
            }
        }
    }
},
```

```
"variations": [
  {
    "customPayload": {
      "value": "string"
    },
    "imageResponseCard": {
      "buttons": [
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
],
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
  "string" : {
    "allowedInputTypes": {
      "allowAudioInput": boolean,
      "allowDTMFInput": boolean
    },
    "allowInterrupt": boolean,
    "audioAndDTMFInputSpecification": {
      "audioSpecification": {
        "endTimeoutMs": number,
        "maxLengthMs": number
      },
      "dtmfSpecification": {
        "deletionCharacter": "string",
        "endCharacter": "string",
        "endTimeoutMs": number,
        "maxLength": number
      },
      "startTimeoutMs": number
    },
    "textInputSpecification": {
      "startTimeoutMs": number
    }
  }
},
"sampleUtterances": [
  {
    "utterance": "string"
  }
],
"waitAndContinueSpecification": {
  "active": boolean,
  "continueResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {

```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    },
    "variations": [
        {
            "customPayload": {
                "value": "string"
            },
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            },
            "plainTextMessage": {
                "value": "string"
            },
            "ssmlMessage": {
                "value": "string"
            }
        }
    ]
},
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
}
},
},
"defaultValueSpecification": {
    "defaultValueList": [
        {
            "defaultValue": "string"
        }
    ]
},
"promptSpecification": {
    "allowInterrupt": boolean,
    "maxRetries": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {

```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
"messageSelectionStrategy": "string",
"promptAttemptsSpecification": {
    "string" : {
        "allowedInputTypes": {
            "allowAudioInput": boolean,
            "allowDTMFInput": boolean
        },
        "allowInterrupt": boolean,
        "audioAndDTMFInputSpecification": {
            "audioSpecification": {
                "endTimeMs": number,
                "maxLengthMs": number
            },
            "dtmfSpecification": {
                "deletionCharacter": "string",
                "endCharacter": "string",
                "endTimeMs": number,
                "maxLength": number
            },
            "startTimeoutMs": number
        },
        "textInputSpecification": {
            "startTimeoutMs": number
        }
    }
},
"sampleUtterances": [
    {
        "utterance": "string"
    }
],
"slotCaptureSetting": {
    "captureConditional": {
        "active": boolean,
        "conditionalBranches": [
            {
                "condition": {
                    "expressionString": "string"
                },
                "name": "string",
                "nextStep": {
                    "dialogAction": {
```

```
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
                            "value": "string"
                        }
                    }
                ]
            }
        ]
    }
}
```

```
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
}
},
"captureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"captureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                }
            }
        }
    ]
}
```

```
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
},
"codeHook": {
    "active": boolean,
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string",
    "postCodeHookSpecification": {
        "failureConditional": {
            "active": boolean,
            "conditionalBranches": [
                {
                    "condition": {
                        "expressionString": "string"
                    },
                    "name": "string",
                    "nextStep": {
                        "dialogAction": {
                            "slotToElicit": "string",
                            "suppressNextMessage": boolean,
                            "type": "string"
                        },
                        "intent": {
                            "name": "string",
                            "slots": {
                                "string" : {
                                    "shape": "string",
                                    "value": {
                                        "interpretedValue": "string"
                                    }
                                },
                                "values": [
                                    "SlotValueOverride"
                                ]
                            }
                        }
                    }
                }
            ]
        }
    }
}
```

```
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ],
    "defaultBranch": {
        "nextStep": {
            "dialogAction": {
                "slotToElicit": "string",
                "type": "string"
            }
        }
    }
}
```

```
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"response": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        "ssmlMessage": {
            "value": "string"
        }
    ]
}
]
}
],
"failureNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"failureResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {

```

```
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      },
      "plainTextMessage": {
        "value": "string"
      },
      "ssmlMessage": {
        "value": "string"
      }
    }
  ]
},
"successConditional": {
  "active": boolean,
  "conditionalBranches": [
    {
      "condition": {
        "expressionString": "string"
      },
      "name": "string",
      "nextStep": {
        "dialogAction": {
          "slotToElicit": "string",
          "suppressNextMessage": boolean,
          "type": "string"
        },
        "intent": {
          "name": "string",
          "slots": {
            "string" : {
              "shape": "string",
              "value": {
                "interpretedValue": "string"
              },
              "values": [
                "SlotValueOverride"
              ]
            }
          }
        },
        "sessionAttributes": {
          "string" : "string"
        }
      }
    },
    "response": {
      "allowInterrupt": boolean,
      "messageGroups": [
        {
          "message": {
            "customPayload": {
              "value": "string"
            },
            "imageResponseCard": {
              "buttons": [
                {
                  "text": "string",
                  "value": "string"
                }
              ]
            }
          }
        }
      ]
    }
  ]
}
```

```
        }
    ],
    "imageUrl": "string",
    "subtitle": "string",
    "title": "string"
},
"plainTextMessage": {
    "value": "string"
},
"ssmlMessage": {
    "value": "string"
}
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"defaultBranch": {
    "nextStep": {
        "dialogAction": {
            "slotToElicit": "string",
            "suppressNextMessage": boolean,
            "type": "string"
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
}
```

```
        },
        "response": {
            "allowInterrupt": boolean,
            "messageGroups": [
                {
                    "message": {
                        "customPayload": {
                            "value": "string"
                        },
                        "imageResponseCard": {
                            "buttons": [
                                {
                                    "text": "string",
                                    "value": "string"
                                }
                            ],
                            "imageUrl": "string",
                            "subtitle": "string",
                            "title": "string"
                        },
                        "plainTextMessage": {
                            "value": "string"
                        },
                        "ssmlMessage": {
                            "value": "string"
                        }
                    },
                    "variations": [
                        {
                            "customPayload": {
                                "value": "string"
                            },
                            "imageResponseCard": {
                                "buttons": [
                                    {
                                        "text": "string",
                                        "value": "string"
                                    }
                                ],
                                "imageUrl": "string",
                                "subtitle": "string",
                                "title": "string"
                            },
                            "plainTextMessage": {
                                "value": "string"
                            },
                            "ssmlMessage": {
                                "value": "string"
                            }
                        }
                    ]
                }
            ],
            "successNextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                },
                "intent": {
                    "name": "string",
                    "slots": {
                        "string" : {

```

```
        "shape": "string",
        "value": {
            "interpretedValue": "string"
        },
        "values": [
            "SlotValueOverride"
        ]
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"successResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "timeoutConditional": {
            "active": boolean,
            "conditionalBranches": [
                {
                    "condition": {
                        "expressionString": "string"
                    },
                    "name": "string",
                    "nextStep": {
                        "dialogAction": {
                            "slotToElicit": "string",
                            "suppressNextMessage": boolean,
                            "type": "string"
                        },
                        "intent": {
                            "name": "string",
                            "slots": {
                                "string" : {
                                    "shape": "string",
                                    "value": {
                                        "interpretedValue": "string"
                                    },
                                    "values": [
                                        "SlotValueOverride"
                                    ]
                                }
                            }
                        }
                    },
                    "sessionAttributes": {
                        "string" : "string"
                    }
                },
                "response": {
                    "allowInterrupt": boolean,
                    "messageGroups": [
                        {
                            "message": {
                                "customPayload": {
                                    "value": "string"
                                },
                                "imageResponseCard": {
                                    "buttons": [
                                        {
                                            "text": "string",
                                            "value": "string"
                                        }
                                    ],
                                    "imageUrl": "string",
                                    "subtitle": "string",
                                    "title": "string"
                                },
                                "plainTextMessage": {
                                    "value": "string"
                                },
                                "ssmlMessage": {
                                    "value": "string"
                                }
                            },
                            "variations": [
                                {
                                    "customPayload": {
                                        "value": "string"
                                    },
                                    "imageResponseCard": {
                                        "buttons": [

```

```

        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
],
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          }
        }
      }
    ]
  }
}

```

```
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
},
"timeoutNextStep": {
    "dialogAction": {
        "slotToElicit": "string",
        "suppressNextMessage": boolean,
        "type": "string"
    },
    "intent": {
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "value": {
                    "interpretedValue": "string"
                },
                "values": [
                    "SlotValueOverride"
                ]
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"timeoutResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
    {
        "message": {
            "customPayload": {
```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
    {
        "customPayload": {
            "value": "string"
        },
        "imageResponseCard": {
            "buttons": [
                {
                    "text": "string",
                    "value": "string"
                }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
]
}
},
"elicitationCodeHook": {
    "enableCodeHookInvocation": boolean,
    "invocationLabel": "string"
},
"failureConditional": {
    "active": boolean,
    "conditionalBranches": [
        {
            "condition": {
                "expressionString": "string"
            },
            "name": "string",
            "nextStep": {
                "dialogAction": {
                    "slotToElicit": "string",
                    "suppressNextMessage": boolean,
                    "type": "string"
                }
            }
        }
    ]
}
}
```

```
        },
        "intent": {
            "name": "string",
            "slots": {
                "string" : {
                    "shape": "string",
                    "value": {
                        "interpretedValue": "string"
                    },
                    "values": [
                        "SlotValueOverride"
                    ]
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    },
    "response": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    }
}
```

```
        }
      ]
    ]
  ]
},
"defaultBranch": {
  "nextStep": {
    "dialogAction": {
      "slotToElicit": "string",
      "suppressNextMessage": boolean,
      "type": "string"
    },
    "intent": {
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": {
            "interpretedValue": "string"
          },
          "values": [
            "SlotValueOverride"
          ]
        }
      }
    },
    "sessionAttributes": {
      "string" : "string"
    }
  },
  "response": {
    "allowInterrupt": boolean,
    "messageGroups": [
      {
        "message": {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [
              {
                "text": "string",
                "value": "string"
              }
            ],
            "imageUrl": "string",
            "subtitle": "string",
            "title": "string"
          }
        },
        "plainTextMessage": {
          "value": "string"
        },
        "ssmlMessage": {
          "value": "string"
        }
      },
      "variations": [
        {
          "customPayload": {
            "value": "string"
          },
          "imageResponseCard": {
            "buttons": [

```

```
        {
          "text": "string",
          "value": "string"
        }
      ],
      "imageUrl": "string",
      "subtitle": "string",
      "title": "string"
    },
    "plainTextMessage": {
      "value": "string"
    },
    "ssmlMessage": {
      "value": "string"
    }
  }
]
}
},
"failureNextStep": {
  "dialogAction": {
    "slotToElicit": "string",
    "suppressNextMessage": boolean,
    "type": "string"
  },
  "intent": {
    "name": "string",
    "slots": {
      "string" : {
        "shape": "string",
        "value": {
          "interpretedValue": "string"
        },
        "values": [
          "SlotValueOverride"
        ]
      }
    }
  },
  "sessionAttributes": {
    "string" : "string"
  }
},
"failureResponse": {
  "allowInterrupt": boolean,
  "messageGroups": [
    {
      "message": {
        "customPayload": {
          "value": "string"
        },
        "imageResponseCard": {
          "buttons": [
            {
              "text": "string",
              "value": "string"
            }
          ],
          "imageUrl": "string",
          "subtitle": "string",
          "title": "string"
        },
        "plainTextMessage": {

```

```
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
},
"variations": [
{
    "customPayload": {
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
"slotConstraint": "string",
"waitAndContinueSpecification": {
    "active": boolean,
    "continueResponse": {
        "allowInterrupt": boolean,
        "messageGroups": [
            {
                "message": {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                },
                "variations": [
                    {
                        "customPayload": {
```

```
        "value": "string"
    },
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    },
    "plainTextMessage": {
        "value": "string"
    },
    "ssmlMessage": {
        "value": "string"
    }
}
]
}
],
},
"stillWaitingResponse": {
    "allowInterrupt": boolean,
    "frequencyInSeconds": number,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    }
                }
            ]
        }
    ]
}
```

```
        },
        "plainTextMessage": {
            "value": "string"
        },
        "ssmlMessage": {
            "value": "string"
        }
    }
}
],
"timeoutInSeconds": number
},
"waitingResponse": {
    "allowInterrupt": boolean,
    "messageGroups": [
        {
            "message": {
                "customPayload": {
                    "value": "string"
                },
                "imageResponseCard": {
                    "buttons": [
                        {
                            "text": "string",
                            "value": "string"
                        }
                    ],
                    "imageUrl": "string",
                    "subtitle": "string",
                    "title": "string"
                },
                "plainTextMessage": {
                    "value": "string"
                },
                "ssmlMessage": {
                    "value": "string"
                }
            },
            "variations": [
                {
                    "customPayload": {
                        "value": "string"
                    },
                    "imageResponseCard": {
                        "buttons": [
                            {
                                "text": "string",
                                "value": "string"
                            }
                        ],
                        "imageUrl": "string",
                        "subtitle": "string",
                        "title": "string"
                    },
                    "plainTextMessage": {
                        "value": "string"
                    },
                    "ssmlMessage": {
                        "value": "string"
                    }
                }
            ]
        }
    ]
}
```

```
    }  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [botId \(p. 653\)](#)

The identifier of the bot that contains the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 653\)](#)

The identifier of the slot version that contains the slot. Will always be DRAFT.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### [creationDateTime \(p. 653\)](#)

The timestamp of the date and time that the slot was created.

Type: Timestamp

### [description \(p. 653\)](#)

The updated description of the bot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

### [intentId \(p. 653\)](#)

The intent that contains the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [lastUpdatedDateTime \(p. 653\)](#)

The timestamp of the date and time that the slot was last updated.

Type: Timestamp

### [localeId \(p. 653\)](#)

The locale that contains the slot.

Type: String

### [multipleValuesSetting \(p. 653\)](#)

Indicates whether the slot accepts multiple values in one response.

Type: [MultipleValuesSetting \(p. 834\)](#) object

### [obfuscationSetting \(p. 653\)](#)

The updated setting that determines whether the slot value is obfuscated in the Amazon CloudWatch logs.

Type: [ObfuscationSetting \(p. 836\)](#) object

### [slotId \(p. 653\)](#)

The unique identifier of the slot that was updated.

Type: String

Length Constraints: Fixed length of 10.

Pattern:  $^[\wedge-9a-zA-Z]+$$

### [slotName \(p. 653\)](#)

The updated name of the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern:  $^([a-zA-Z][\wedge-]{0,99})+$$

### [slotTypeId \(p. 653\)](#)

The updated identifier of the slot type that provides values for the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern:  $^((AMAZON\.)[a-zA-Z_]+|\w{1,24})+$$

### [subSlotSetting \(p. 653\)](#)

Specifications for the constituent sub slots and the expression for the composite slot.

Type: [SubSlotSetting \(p. 881\)](#) object

### [valueElicitationSetting \(p. 653\)](#)

The updated prompts that Amazon Lex sends to the user to elicit a response that provides a value for the slot.

Type: [SlotValueElicitationSetting \(p. 873\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

**InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

**PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

**ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

**ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

**ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateSlotType

Service: Amazon Lex Model Building V2

Updates the configuration of an existing slot type.

### Request Syntax

```
PUT /bots/botId/botversions/botVersion/botlocales/localeId/slottypes/slotTypeId/ HTTP/1.1
Content-type: application/json

{
    "compositeSlotTypeSetting": {
        "subSlots": [
            {
                "name": "string",
                "slotTypeId": "string"
            }
        ]
    },
    "description": "string",
    "externalSourceSetting": {
        "grammarSlotTypeSetting": {
            "source": {
                "kmsKeyArn": "string",
                "s3BucketName": "string",
                "s3ObjectKey": "string"
            }
        }
    },
    "parentSlotTypeSignature": "string",
    "slotTypeName": "string",
    "slotTypeValues": [
        {
            "sampleValue": {
                "value": "string"
            },
            "synonyms": [
                {
                    "value": "string"
                }
            ]
        }
    ],
    "valueSelectionSetting": {
        "advancedRecognitionSetting": {
            "audioRecognitionStrategy": "string"
        },
        "regexFilter": {
            "pattern": "string"
        },
        "resolutionStrategy": "string"
    }
}
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botId (p. 683)**

The identifier of the bot that contains the slot type.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion (p. 683)**

The version of the bot that contains the slot type. Must be DRAFT.

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId (p. 683)**

The identifier of the language and locale that contains the slot type. The string must match one of the supported locales. For more information, see [Supported languages](#).

Required: Yes

#### **slotTypeId (p. 683)**

The unique identifier of the slot type to update.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

## **Request Body**

The request accepts the following data in JSON format.

#### **compositeSlotTypeSetting (p. 683)**

Specifications for a composite slot type.

Type: [CompositeSlotTypeSetting \(p. 777\)](#) object

Required: No

#### **description (p. 683)**

The new description of the slot type.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **externalSourceSetting (p. 683)**

Provides information about the external source of the slot type's definition.

Type: [ExternalSourceSetting \(p. 802\)](#) object

Required: No

#### **parentSlotTypeSignature (p. 683)**

The new built-in slot type that should be used as the parent of this slot type.

Type: String

Required: No

#### [slotTypeName \(p. 683\)](#)

The new name of the slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### [slotTypeValues \(p. 683\)](#)

A new list of values and their optional synonyms that define the values that the slot type can take.

Type: Array of [SlotTypeValue \(p. 871\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10000 items.

Required: No

#### [valueSelectionSetting \(p. 683\)](#)

The strategy that Amazon Lex should use when deciding on a value from the list of slot type values.

Type: [SlotValueSelectionSetting \(p. 877\)](#) object

Required: No

## Response Syntax

```
HTTP/1.1 202
Content-type: application/json

{
    "botId": "string",
    "botVersion": "string",
    "compositeSlotTypeSetting": {
        "subSlots": [
            {
                "name": "string",
                "slotTypeId": "string"
            }
        ]
    },
    "creationDateTime": number,
    "description": "string",
    "externalSourceSetting": {
        "grammarSlotTypeSetting": {
            "source": {
                "kmsKeyArn": "string",
                "s3BucketName": "string",
                "s3ObjectKey": "string"
            }
        }
    },
    "lastUpdatedDateTime": number,
    "localeId": "string",
    "parentSlotTypeSignature": "string",
```

```
"slotTypeId": "string",
"slotTypeName": "string",
"slotTypeValues": [
  {
    "sampleValue": {
      "value": "string"
    },
    "synonyms": [
      {
        "value": "string"
      }
    ]
  }
],
"valueSelectionSetting": {
  "advancedRecognitionSetting": {
    "audioRecognitionStrategy": "string"
  },
  "regexFilter": {
    "pattern": "string"
  },
  "resolutionStrategy": "string"
}
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 202 response.

The following data is returned in JSON format by the service.

### [botId \(p. 685\)](#)

The identifier of the bot that contains the slot type.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### [botVersion \(p. 685\)](#)

The version of the bot that contains the slot type. This is always DRAFT.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

### [compositeSlotTypeSetting \(p. 685\)](#)

Specifications for a composite slot type.

Type: [CompositeSlotTypeSetting \(p. 777\)](#) object

### [creationDateTime \(p. 685\)](#)

The timestamp of the date and time that the slot type was created.

Type: Timestamp

### [description \(p. 685\)](#)

The updated description of the slot type.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

[externalSourceSetting \(p. 685\)](#)

Provides information about the external source of the slot type's definition.

Type: [ExternalSourceSetting \(p. 802\)](#) object

[lastUpdatedDateTime \(p. 685\)](#)

A timestamp of the date and time that the slot type was last updated.

Type: Timestamp

[localeId \(p. 685\)](#)

The language and locale of the updated slot type.

Type: String

[parentSlotTypeSignature \(p. 685\)](#)

The updated signature of the built-in slot type that is the parent of this slot type.

Type: String

[slotTypeId \(p. 685\)](#)

The unique identifier of the updated slot type.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

[slotTypeName \(p. 685\)](#)

The updated name of the slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

[slotTypeValues \(p. 685\)](#)

The updated values that the slot type provides.

Type: Array of [SlotTypeValue \(p. 871\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10000 items.

[valueSelectionSetting \(p. 685\)](#)

The updated strategy that Amazon Lex uses to determine which value to select from the slot type.

Type: [SlotValueSelectionSetting \(p. 877\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **ConflictException**

The action that you tried to perform couldn't be completed because the resource is in a conflicting state. For example, deleting a bot that is in the CREATING state. Try your request again.

HTTP Status Code: 409

### **InternalServerException**

The service encountered an unexpected condition. Try your request again.

HTTP Status Code: 500

### **PreconditionFailedException**

Your request couldn't be completed because one or more request fields aren't valid. Check the fields in your request and try again.

HTTP Status Code: 412

### **ServiceQuotaExceededException**

You have reached a quota for your bot.

HTTP Status Code: 402

### **ThrottlingException**

Your request rate is too high. Reduce the frequency of requests.

HTTP Status Code: 429

### **ValidationException**

One of the input parameters in your request isn't valid. Check the parameters and try your request again.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## Amazon Lex Runtime V2

The following actions are supported by Amazon Lex Runtime V2:

- [DeleteSession \(p. 690\)](#)
- [GetSession \(p. 693\)](#)

- [PutSession \(p. 698\)](#)
- [RecognizeText \(p. 703\)](#)
- [RecognizeUtterance \(p. 710\)](#)
- [StartConversation \(p. 717\)](#)

## DeleteSession

Service: Amazon Lex Runtime V2

Removes session information for a specified bot, alias, and user ID.

You can use this operation to restart a conversation with a bot. When you remove a session, the entire history of the session is removed so that you can start again.

You don't need to delete a session. Sessions have a time limit and will expire. Set the session time limit when you create the bot. The default is 5 minutes, but you can specify anything between 1 minute and 24 hours.

If you specify a bot or alias ID that doesn't exist, you receive a `BadRequestException`.

If the locale doesn't exist in the bot, or if the locale hasn't been enables for the alias, you receive a `BadRequestException`.

### Request Syntax

```
DELETE /bots/botId/botAliases/botAliasId/botLocales/localeId/sessions/sessionId HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botAliasId** (p. 690)

The alias identifier in use for the bot that contains the session data.

Required: Yes

#### **botId** (p. 690)

The identifier of the bot that contains the session data.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **localeId** (p. 690)

The locale where the session is in use.

Length Constraints: Minimum length of 1.

Required: Yes

#### **sessionId** (p. 690)

The identifier of the session to delete.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: Yes

### Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "botAliasId": "string",
    "botId": "string",
    "localeId": "string",
    "sessionId": "string"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **botAliasId** (p. 691)

The alias identifier in use for the bot that contained the session data.

Type: String

### **botId** (p. 691)

The identifier of the bot that contained the session data.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

### **localeId** (p. 691)

The locale where the session was used.

Type: String

Length Constraints: Minimum length of 1.

### **sessionId** (p. 691)

The identifier of the deleted session.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **AccessDeniedException**

HTTP Status Code: 403

### **ConflictException**

HTTP Status Code: 409

**InternalServerException**

HTTP Status Code: 500

**ResourceNotFoundException**

HTTP Status Code: 404

**ThrottlingException**

HTTP Status Code: 429

**ValidationException**

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## GetSession

Service: Amazon Lex Runtime V2

Returns session information for a specified bot, alias, and user.

For example, you can use this operation to retrieve session information for a user that has left a long-running session in use.

If the bot, alias, or session identifier doesn't exist, Amazon Lex V2 returns a `BadRequestException`. If the locale doesn't exist or is not enabled for the alias, you receive a `BadRequestException`.

### Request Syntax

```
GET /bots/botId/botAliases/botAliasId/botLocales/localeId/sessions/sessionId HTTP/1.1
```

### URI Request Parameters

The request uses the following URI parameters.

#### **botAliasId** (p. 693)

The alias identifier in use for the bot that contains the session data.

Required: Yes

#### **botId** (p. 693)

The identifier of the bot that contains the session data.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **localeId** (p. 693)

The locale where the session is in use.

Length Constraints: Minimum length of 1.

Required: Yes

#### **sessionId** (p. 693)

The identifier of the session to return.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: Yes

### Request Body

The request does not have a request body.

### Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{  
    "interpretations": [  
        {  
            "intent": {  
                "confirmationState": "string",  
                "name": "string",  
                "slots": {  
                    "string" : {  
                        "shape": "string",  
                        "subSlots": {  
                            "string" : "Slot"  
                        },  
                        "value": {  
                            "interpretedValue": "string",  
                            "originalValue": "string",  
                            "resolvedValues": [ "string" ]  
                        },  
                        "values": [  
                            "Slot"  
                        ]  
                    }  
                },  
                "state": "string"  
            },  
            "nluConfidence": {  
                "score": number  
            },  
            "sentimentResponse": {  
                "sentiment": "string",  
                "sentimentScore": {  
                    "mixed": number,  
                    "negative": number,  
                    "neutral": number,  
                    "positive": number  
                }  
            }  
        }  
    ],  
    "messages": [  
        {  
            "content": "string",  
            "contentType": "string",  
            "imageResponseCard": {  
                "buttons": [  
                    {  
                        "text": "string",  
                        "value": "string"  
                    }  
                ],  
                "imageUrl": "string",  
                "subtitle": "string",  
                "title": "string"  
            }  
        }  
    ],  
    "sessionId": "string",  
    "sessionState": {  
        "activeContexts": [  
            {  
                "contextAttributes": {  
                    "string" : "string"  
                },  
                "name": "string",  
                "timeToLive": {  
                    "timeToLiveInSeconds": number,  
                    "turnsToLive": number  
                }  
            }  
        ]  
    }  
}
```

```
        }
    ],
    "dialogAction": {
        "slotElicitationStyle": "string",
        "slotToElicit": "string",
        "subSlotToElicit": {
            "name": "string",
            "subSlotToElicit": "ElicitSubSlot"
        },
        "type": "string"
    },
    "intent": {
        "confirmationState": "string",
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "subSlots": {
                    "string" : "Slot"
                },
                "value": {
                    "interpretedValue": "string",
                    "originalValue": "string",
                    "resolvedValues": [ "string" ]
                },
                "values": [
                    "Slot"
                ]
            }
        },
        "state": "string"
    },
    "originatingRequestId": "string",
    "runtimeHints": {
        "slotHints": {
            "string" : {
                "string" : {
                    "runtimeHintValues": [
                        {
                            "phrase": "string"
                        }
                    ],
                    "subSlotHints": {
                        "string" : "RuntimeHintDetails"
                    }
                }
            }
        },
        "sessionAttributes": {
            "string" : "string"
        }
    }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [interpretations \(p. 693\)](#)

A list of intents that Amazon Lex V2 determined might satisfy the user's utterance.

Each interpretation includes the intent, a score that indicates how confident Amazon Lex V2 is that the interpretation is the correct one, and an optional sentiment response that indicates the sentiment expressed in the utterance.

Type: Array of [Interpretation \(p. 914\)](#) objects

Array Members: Maximum number of 5 items.

#### **messages (p. 693)**

A list of messages that were last sent to the user. The messages are ordered based on the order that you returned the messages from your Lambda function or the order that messages are defined in the bot.

Type: Array of [Message \(p. 915\)](#) objects

Array Members: Maximum number of 10 items.

#### **sessionId (p. 693)**

The identifier of the returned session.

Type: String

Length Constraints: Minimum length of 1.

#### **sessionState (p. 693)**

Represents the current state of the dialog between the user and the bot.

You can use this to determine the progress of the conversation and what the next action might be.

Type: [SessionState \(p. 923\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **AccessDeniedException**

HTTP Status Code: 403

### **InternalServerError**

HTTP Status Code: 500

### **ResourceNotFoundException**

HTTP Status Code: 404

### **ThrottlingException**

HTTP Status Code: 429

### **ValidationException**

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## PutSession

Service: Amazon Lex Runtime V2

Creates a new session or modifies an existing session with an Amazon Lex V2 bot. Use this operation to enable your application to set the state of the bot.

### Request Syntax

```
POST /bots/botId/botAliases/botAliasId/botLocales/localeId/sessions/sessionId HTTP/1.1
ResponseContentType: responseContentType
Content-type: application/json

{
  "messages": [
    {
      "content": "string",
      "contentType": "string",
      "imageResponseCard": {
        "buttons": [
          {
            "text": "string",
            "value": "string"
          }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
      }
    }
  ],
  "requestAttributes": {
    "string": "string"
  },
  "sessionState": {
    "activeContexts": [
      {
        "contextAttributes": {
          "string": "string"
        },
        "name": "string",
        "timeToLive": {
          "timeToLiveInSeconds": number,
          "turnsToLive": number
        }
      }
    ],
    "dialogAction": {
      "slotElicitationStyle": "string",
      "slotToElicit": "string",
      "subSlotToElicit": {
        "name": "string",
        "subSlotToElicit": "ElicitSubSlot"
      },
      "type": "string"
    },
    "intent": {
      "confirmationState": "string",
      "name": "string",
      "slots": [
        "string: {
          "shape": "string",
          "subSlots": [
            "string": "Slot"
          ],
        }
      ]
    }
  }
}
```

```
        "value": {
            "interpretedValue": "string",
            "originalValue": "string",
            "resolvedValues": [ "string" ]
        },
        "values": [
            "Slot"
        ]
    }
},
"state": "string"
},
"originatingRequestId": "string",
"runtimeHints": {
    "slotHints": {
        "string": {
            "string": {
                "runtimeHintValues": [
                    {
                        "phrase": "string"
                    }
                ],
                "subSlotHints": {
                    "string": "RuntimeHintDetails"
                }
            }
        }
    }
},
"sessionAttributes": {
    "string": "string"
}
}
```

## URI Request Parameters

The request uses the following URI parameters.

### **botAliasId (p. 698)**

The alias identifier of the bot that receives the session data.

Required: Yes

### **botId (p. 698)**

The identifier of the bot that receives the session data.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **localeId (p. 698)**

The locale where the session is in use.

Length Constraints: Minimum length of 1.

Required: Yes

### **responseContentType (p. 698)**

The message that Amazon Lex V2 returns in the response can be either text or speech depending on the value of this parameter.

- If the value is `text/plain; charset=utf-8`, Amazon Lex V2 returns text in the response.

Length Constraints: Minimum length of 1.

#### [sessionId \(p. 698\)](#)

The identifier of the session that receives the session data.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: `[0-9a-zA-Z._:-]+`

Required: Yes

## Request Body

The request accepts the following data in JSON format.

#### [messages \(p. 698\)](#)

A list of messages to send to the user. Messages are sent in the order that they are defined in the list.

Type: Array of [Message \(p. 915\)](#) objects

Array Members: Maximum number of 10 items.

Required: No

#### [requestAttributes \(p. 698\)](#)

Request-specific information passed between Amazon Lex V2 and the client application.

The namespace `x-amz-lex:` is reserved for special attributes. Don't create any request attributes with the prefix `x-amz-lex:`.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

#### [sessionState \(p. 698\)](#)

Sets the state of the session with the user. You can use this to set the current intent, attributes, context, and dialog action. Use the dialog action to determine the next step that Amazon Lex V2 should use in the conversation with the user.

Type: [SessionState \(p. 923\)](#) object

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-Type: contentType
x-amz-lex-messages: messages
x-amz-lex-session-state: sessionState
x-amz-lex-request-attributes: requestAttributes
x-amz-lex-session-id: sessionId

audioStream
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

### **contentType** (p. 700)

The type of response. Same as the type specified in the `responseContentType` field in the request.

Length Constraints: Minimum length of 1.

### **messages** (p. 700)

A list of messages that were last sent to the user. The messages are ordered based on how you return the messages from your Lambda function or the order that the messages are defined in the bot.

Length Constraints: Minimum length of 1.

### **requestAttributes** (p. 700)

Request-specific information passed between the client application and Amazon Lex V2. These are the same as the `requestAttribute` parameter in the call to the PutSession operation.

Length Constraints: Minimum length of 1.

### **sessionId** (p. 700)

The identifier of the session that received the data.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

### **sessionState** (p. 700)

Represents the current state of the dialog between the user and the bot.

Use this to determine the progress of the conversation and what the next action may be.

Length Constraints: Minimum length of 1.

The response returns the following as the HTTP body.

### **audioStream** (p. 700)

If the requested content type was audio, the audio version of the message to convey to the user.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **AccessDeniedException**

HTTP Status Code: 403

### **BadGatewayException**

HTTP Status Code: 502

### **ConflictException**

HTTP Status Code: 409

### **DependencyFailedException**

HTTP Status Code: 424

### **InternalServerError**

HTTP Status Code: 500

### **ResourceNotFoundException**

HTTP Status Code: 404

### **ThrottlingException**

HTTP Status Code: 429

### **ValidationException**

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## RecognizeText

Service: Amazon Lex Runtime V2

Sends user input to Amazon Lex V2. Client applications use this API to send requests to Amazon Lex V2 at runtime. Amazon Lex V2 then interprets the user input using the machine learning model that it build for the bot.

In response, Amazon Lex V2 returns the next message to convey to the user and an optional response card to display.

If the optional post-fulfillment response is specified, the messages are returned as follows. For more information, see [PostFulfillmentStatusSpecification](#).

- **Success message** - Returned if the Lambda function completes successfully and the intent state is fulfilled or ready fulfillment if the message is present.
- **Failed message** - The failed message is returned if the Lambda function throws an exception or if the Lambda function returns a failed intent state without a message.
- **Timeout message** - If you don't configure a timeout message and a timeout, and the Lambda function doesn't return within 30 seconds, the timeout message is returned. If you configure a timeout, the timeout message is returned when the period times out.

For more information, see [Completion message](#).

## Request Syntax

```
POST /bots/botId/botAliases/botAliasId/botLocales/localeId/sessions/sessionId/text HTTP/1.1
Content-type: application/json

{
  "requestAttributes": {
    "string" : "string"
  },
  "sessionState": {
    "activeContexts": [
      {
        "contextAttributes": {
          "string" : "string"
        },
        "name": "string",
        "timeToLive": {
          "timeToLiveInSeconds": number,
          "turnsToLive": number
        }
      }
    ],
    "dialogAction": {
      "slotElicitationStyle": "string",
      "slotToElicit": "string",
      "subSlotToElicit": {
        "name": "string",
        "subSlotToElicit": "ElicitSubSlot"
      },
      "type": "string"
    },
    "intent": {
      "confirmationState": "string",
      "name": "string",
      "slots": {
        "string" : {
          "shape": "string",
          "value": "string"
        }
      }
    }
  }
}
```

```
        "subSlots": {
            "string" : "Slot"
        },
        "value": {
            "interpretedValue": "string",
            "originalValue": "string",
            "resolvedValues": [ "string" ]
        },
        "values": [
            "Slot"
        ]
    }
},
"state": "string"
},
"originatingRequestId": "string",
"runtimeHints": {
    "slotHints": {
        "string" : {
            "string" : {
                "runtimeHintValues": [
                    {
                        "phrase": "string"
                    }
                ],
                "subSlotHints": {
                    "string" : "RuntimeHintDetails"
                }
            }
        }
    }
},
"sessionAttributes": {
    "string" : "string"
}
},
"text": "string"
}
```

## URI Request Parameters

The request uses the following URI parameters.

### [botAliasId \(p. 703\)](#)

The alias identifier in use for the bot that processes the request.

Required: Yes

### [botId \(p. 703\)](#)

The identifier of the bot that processes the request.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### [localeId \(p. 703\)](#)

The locale where the session is in use.

Length Constraints: Minimum length of 1.

Required: Yes

#### [sessionId \(p. 703\)](#)

The identifier of the user session that is having the conversation.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: Yes

## Request Body

The request accepts the following data in JSON format.

#### [requestAttributes \(p. 703\)](#)

Request-specific information passed between the client application and Amazon Lex V2

The namespace x-amz-lex : is reserved for special attributes. Don't create any request attributes with the prefix x-amz-lex :.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

#### [sessionState \(p. 703\)](#)

The current state of the dialog between the user and the bot.

Type: [SessionState \(p. 923\)](#) object

Required: No

#### [text \(p. 703\)](#)

The text that the user entered. Amazon Lex V2 interprets this text.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "interpretations": [
    {
      "intent": {
        "confirmationState": "string",
        "name": "string",
        "slots": {
          "string" : {
            "shape": "string",
            "subSlots": {

```

```
        "string" : "Slot"
    },
    "value": {
        "interpretedValue": "string",
        "originalValue": "string",
        "resolvedValues": [ "string" ]
    },
    "values": [
        "Slot"
    ]
},
"state": "string"
},
"nluConfidence": {
    "score": number
},
"sentimentResponse": {
    "sentiment": "string",
    "sentimentScore": {
        "mixed": number,
        "negative": number,
        "neutral": number,
        "positive": number
    }
}
}
],
"messages": [
{
    "content": "string",
    "contentType": "string",
    "imageResponseCard": {
        "buttons": [
            {
                "text": "string",
                "value": "string"
            }
        ],
        "imageUrl": "string",
        "subtitle": "string",
        "title": "string"
    }
}
],
"requestAttributes": {
    "string" : "string"
},
"sessionId": "string",
"sessionState": {
    "activeContexts": [
        {
            "contextAttributes": {
                "string" : "string"
            },
            "name": "string",
            "timeToLive": {
                "timeToLiveInSeconds": number,
                "turnsToLive": number
            }
        }
    ],
    "dialogAction": {
        "slotElicitationStyle": "string",
        "slotToElicit": "string",
        "subSlotToElicit": {

```

```
        "name": "string",
        "subSlotToElicit": "ElicitSubSlot"
    },
    "type": "string"
},
"intent": {
    "confirmationState": "string",
    "name": "string",
    "slots": {
        "string" : {
            "shape": "string",
            "subSlots": {
                "string" : "Slot"
            },
            "value": {
                "interpretedValue": "string",
                "originalValue": "string",
                "resolvedValues": [ "string" ]
            },
            "values": [
                "Slot"
            ]
        }
    },
    "state": "string"
},
"originatingRequestId": "string",
"runtimeHints": {
    "slotHints": {
        "string" : {
            "string" : {
                "runtimeHintValues": [
                    {
                        "phrase": "string"
                    }
                ],
                "subSlotHints": {
                    "string" : "RuntimeHintDetails"
                }
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
}
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [interpretations \(p. 705\)](#)

A list of intents that Amazon Lex V2 determined might satisfy the user's utterance.

Each interpretation includes the intent, a score that indicates how confident Amazon Lex V2 is that the interpretation is the correct one, and an optional sentiment response that indicates the sentiment expressed in the utterance.

Type: Array of [Interpretation \(p. 914\)](#) objects

Array Members: Maximum number of 5 items.

#### **messages (p. 705)**

A list of messages last sent to the user. The messages are ordered based on the order that you returned the messages from your Lambda function or the order that the messages are defined in the bot.

Type: Array of [Message \(p. 915\)](#) objects

Array Members: Maximum number of 10 items.

#### **requestAttributes (p. 705)**

The attributes sent in the request.

Type: String to string map

Key Length Constraints: Minimum length of 1.

#### **sessionId (p. 705)**

The identifier of the session in use.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

#### **sessionState (p. 705)**

Represents the current state of the dialog between the user and the bot.

Use this to determine the progress of the conversation and what the next action may be.

Type: [SessionState \(p. 923\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **AccessDeniedException**

HTTP Status Code: 403

### **BadGatewayException**

HTTP Status Code: 502

### **ConflictException**

HTTP Status Code: 409

### **DependencyFailedException**

HTTP Status Code: 424

### **InternalServerError**

HTTP Status Code: 500

### **ResourceNotFoundException**

HTTP Status Code: 404

### **ThrottlingException**

HTTP Status Code: 429

### **ValidationException**

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## RecognizeUtterance

Service: Amazon Lex Runtime V2

Sends user input to Amazon Lex V2. You can send text or speech. Clients use this API to send text and audio requests to Amazon Lex V2 at runtime. Amazon Lex V2 interprets the user input using the machine learning model built for the bot.

The following request fields must be compressed with gzip and then base64 encoded before you send them to Amazon Lex V2.

- `requestAttributes`
- `sessionState`

The following response fields are compressed using gzip and then base64 encoded by Amazon Lex V2. Before you can use these fields, you must decode and decompress them.

- `inputTranscript`
- `interpretations`
- `messages`
- `requestAttributes`
- `sessionState`

The example contains a Java application that compresses and encodes a Java object to send to Amazon Lex V2, and a second that decodes and decompresses a response from Amazon Lex V2.

If the optional post-fulfillment response is specified, the messages are returned as follows. For more information, see [PostFulfillmentStatusSpecification](#).

- **Success message** - Returned if the Lambda function completes successfully and the intent state is fulfilled or ready fulfillment if the message is present.
- **Failed message** - The failed message is returned if the Lambda function throws an exception or if the Lambda function returns a failed intent state without a message.
- **Timeout message** - If you don't configure a timeout message and a timeout, and the Lambda function doesn't return within 30 seconds, the timeout message is returned. If you configure a timeout, the timeout message is returned when the period times out.

For more information, see [Completion message](#).

### Request Syntax

```
POST /bots/botId/botAliases/botAliasId/botLocales/localeId/sessions/sessionId/utterance
HTTP/1.1
x-amz-lex-session-state: sessionState
x-amz-lex-request-attributes: requestAttributes
Content-Type: requestContentType
Response-Content-Type: responseContentType

inputStream
```

### URI Request Parameters

The request uses the following URI parameters.

### [botAliasId \(p. 710\)](#)

The alias identifier in use for the bot that should receive the request.

Required: Yes

### [botId \(p. 710\)](#)

The identifier of the bot that should receive the request.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### [localeId \(p. 710\)](#)

The locale where the session is in use.

Length Constraints: Minimum length of 1.

Required: Yes

### [requestAttributes \(p. 710\)](#)

Request-specific information passed between the client application and Amazon Lex V2

The namespace x-amz-lex : is reserved for special attributes. Don't create any request attributes for prefix x-amz-lex: .

The `requestAttributes` field must be compressed using gzip and then base64 encoded before sending to Amazon Lex V2.

### [requestContentType \(p. 710\)](#)

Indicates the format for audio input or that the content is text. The header must start with one of the following prefixes:

- PCM format, audio data must be in little-endian byte order.
  - `audio/l16; rate=16000; channels=1`
  - `audio/x-l16; sample-rate=16000; channel-count=1`
  - `audio/lpcm; sample-rate=8000; sample-size-bits=16; channel-count=1; is-big-endian=false`
- Opus format
  - `audio/x-cbr-opus-with-preamble;preamble-size=0;bit-rate=256000;frame-size-milliseconds=4`
- Text format
  - `text/plain; charset=utf-8`

Length Constraints: Minimum length of 1.

Required: Yes

### [responseContentType \(p. 710\)](#)

The message that Amazon Lex V2 returns in the response can be either text or speech based on the `responseContentType` value.

- If the value is `text/plain; charset=utf-8`, Amazon Lex V2 returns text in the response.
- If the value begins with `audio/`, Amazon Lex V2 returns speech in the response. Amazon Lex V2 uses Amazon Polly to generate the speech using the configuration that you specified in the `responseContentType` parameter. For example, if you specify `audio/mpeg` as the value, Amazon Lex V2 returns speech in the MPEG format.

- If the value is audio/pcm, the speech returned is audio/pcm at 16 KHz in 16-bit, little-endian format.
- The following are the accepted values:
  - audio/mpeg
  - audio/ogg
  - audio/pcm (16 KHz)
  - audio/\* (defaults to mpeg)
  - text/plain; charset=utf-8

Length Constraints: Minimum length of 1.

#### [sessionId \(p. 710\)](#)

The identifier of the session in use.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: Yes

#### [sessionState \(p. 710\)](#)

Sets the state of the session with the user. You can use this to set the current intent, attributes, context, and dialog action. Use the dialog action to determine the next step that Amazon Lex V2 should use in the conversation with the user.

The sessionState field must be compressed using gzip and then base64 encoded before sending to Amazon Lex V2.

## Request Body

The request accepts the following binary data.

#### [inputStream \(p. 710\)](#)

User input in PCM or Opus audio format or text format as described in the `requestContentType` parameter.

## Response Syntax

```
HTTP/1.1 200
x-amz-lex-input-mode: inputMode
Content-Type: contentType
x-amz-lex-messages: messages
x-amz-lex-interpretations: interpretations
x-amz-lex-session-state: sessionState
x-amz-lex-request-attributes: requestAttributes
x-amz-lex-session-id: sessionId
x-amz-lex-input-transcript: inputTranscript

audioStream
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

### [contentType \(p. 712\)](#)

Content type as specified in the `responseContentType` in the request.

Length Constraints: Minimum length of 1.

### [inputMode \(p. 712\)](#)

Indicates whether the input mode to the operation was text or speech.

Length Constraints: Minimum length of 1.

### [inputTranscript \(p. 712\)](#)

The text used to process the request.

If the input was an audio stream, the `inputTranscript` field contains the text extracted from the audio stream. This is the text that is actually processed to recognize intents and slot values. You can use this information to determine if Amazon Lex V2 is correctly processing the audio that you send.

The `inputTranscript` field is compressed with gzip and then base64 encoded. Before you can use the contents of the field, you must decode and decompress the contents. See the example for a simple function to decode and decompress the contents.

Length Constraints: Minimum length of 1.

### [interpretations \(p. 712\)](#)

A list of intents that Amazon Lex V2 determined might satisfy the user's utterance.

Each interpretation includes the intent, a score that indicates how confident Amazon Lex V2 is that the interpretation is the correct one, and an optional sentiment response that indicates the sentiment expressed in the utterance.

The `interpretations` field is compressed with gzip and then base64 encoded. Before you can use the contents of the field, you must decode and decompress the contents. See the example for a simple function to decode and decompress the contents.

Length Constraints: Minimum length of 1.

### [messages \(p. 712\)](#)

A list of messages that were last sent to the user. The messages are ordered based on the order that you returned the messages from your Lambda function or the order that the messages are defined in the bot.

The `messages` field is compressed with gzip and then base64 encoded. Before you can use the contents of the field, you must decode and decompress the contents. See the example for a simple function to decode and decompress the contents.

Length Constraints: Minimum length of 1.

### [requestAttributes \(p. 712\)](#)

The attributes sent in the request.

The `requestAttributes` field is compressed with gzip and then base64 encoded. Before you can use the contents of the field, you must decode and decompress the contents.

Length Constraints: Minimum length of 1.

### [sessionId \(p. 712\)](#)

The identifier of the session in use.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

### [sessionState \(p. 712\)](#)

Represents the current state of the dialog between the user and the bot.

Use this to determine the progress of the conversation and what the next action might be.

The `sessionState` field is compressed with gzip and then base64 encoded. Before you can use the contents of the field, you must decode and decompress the contents. See the example for a simple function to decode and decompress the contents.

Length Constraints: Minimum length of 1.

The response returns the following as the HTTP body.

### [audioStream \(p. 712\)](#)

The prompt or statement to send to the user. This is based on the bot configuration and context. For example, if Amazon Lex V2 did not understand the user intent, it sends the `clarificationPrompt` configured for the bot. If the intent requires confirmation before taking the fulfillment action, it sends the `confirmationPrompt`. Another example: Suppose that the Lambda function successfully fulfilled the intent, and sent a message to convey to the user. Then Amazon Lex V2 sends that message in the response.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **AccessDeniedException**

HTTP Status Code: 403

### **BadGatewayException**

HTTP Status Code: 502

### **ConflictException**

HTTP Status Code: 409

### **DependencyFailedException**

HTTP Status Code: 424

### **InternalServerError**

HTTP Status Code: 500

### **ResourceNotFoundException**

HTTP Status Code: 404

### **ThrottlingException**

HTTP Status Code: 429

### ValidationException

HTTP Status Code: 400

## Examples

### Encode and decode a field

The following example provides two functions, one to compress a string with gzip and then base64 encode it and one to decode and decompress a string.

```
package com.amazonaws.deepsense.util;

import com.amazonaws.deepsense.runtime.conversation.serialize.RuntimeSdkSerializer;
import com.fasterxml.jackson.annotation.JsonInclude;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.DeserializationFeature;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.SerializationFeature;
import com.google.common.base.Preconditions;

import java.io.IOException;
import java.util.Base64;

public class CompressionAndEncoding {

    private CompressionAndEncoding() {
    }

    /**
     * Given a generic object
     * 1. Serialize the object using jackson
     * 2. Compress the serialized object
     * 3. Base64 encode the compressed data
     */
    public static String compressAndEncodeBase64(Object object) {
        if (object == null) {
            return null;
        }

        String objectAsString = GsonSerializer.serialize(object);
        byte[] compressedString = Compressions.compress(objectAsString);
        return Base64.getEncoder().encodeToString(compressedString);
    }

    /**
     * Given a base64 encoded, compressed, serialized object
     * 1. Base64 decodes the data
     * 2. Decompresses the base64 decoded data
     * 3. Converts the serialized object into a proper POJO
     */
    public static <T> T decodeBase64AndDecompress(String objectAsString, TypeReference<T>
typeRef) {
        if (objectAsString == null) {
            return null;
        }

        Preconditions.checkNotNull(typeRef, "Serialization class can't be null.");
        byte[] decodedBytes = Base64.getDecoder().decode(objectAsString);
    }
}
```

```
        String decompressedObjectAsString = Compressions.uncompress(decodedBytes);
        try {
            return RuntimeSdkSerializer.instance().readValue(decompressedObjectAsString,
typeRef);
        } catch (IOException e) {
            throw new RuntimeException(String.format("Unable to deserialize string %s",
decompressedObjectAsString), e);
        }
    }
}
```

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## StartConversation

Service: Amazon Lex Runtime V2

Starts an HTTP/2 bidirectional event stream that enables you to send audio, text, or DTMF input in real time. After your application starts a conversation, users send input to Amazon Lex V2 as a stream of events. Amazon Lex V2 processes the incoming events and responds with streaming text or audio events.

Audio input must be in the following format: `audio/lpcm sample-rate=8000 sample-size-bits=16 channel-count=1; is-big-endian=false`.

If the optional post-fulfillment response is specified, the messages are returned as follows. For more information, see [PostFulfillmentStatusSpecification](#).

- **Success message** - Returned if the Lambda function completes successfully and the intent state is fulfilled or ready fulfillment if the message is present.
- **Failed message** - The failed message is returned if the Lambda function throws an exception or if the Lambda function returns a failed intent state without a message.
- **Timeout message** - If you don't configure a timeout message and a timeout, and the Lambda function doesn't return within 30 seconds, the timeout message is returned. If you configure a timeout, the timeout message is returned when the period times out.

For more information, see [Completion message](#).

If the optional update message is configured, it is played at the specified frequency while the Lambda function is running and the update message state is active. If the fulfillment update message is not active, the Lambda function runs with a 30 second timeout.

For more information, see [Update message](#)

The StartConversation operation is supported only in the following SDKs:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Request Syntax

```
POST /bots/botId/botAliases/botAliasId/botLocales/localeId/sessions/sessionId/conversation
HTTP/2
x-amz-lex-conversation-mode: conversationMode
Content-type: application/json

{
  "requestEventStream": {
    "AudioInputEvent": {
      "audioChunk": blob,
      "clientTimestampMillis": number,
      "contentType": "string",
      "eventId": "string"
    },
    "ConfigurationEvent": {
      "clientTimestampMillis": number,
      "disablePlayback": boolean,
      "eventId": "string",
      "requestAttributes": {
        "string": "string"
      }
    }
}
```

```
"responseContentType": "string",
"sessionState": {
    "activeContexts": [
        {
            "contextAttributes": {
                "string" : "string"
            },
            "name": "string",
            "timeToLive": {
                "timeToLiveInSeconds": number,
                "turnsToLive": number
            }
        }
    ],
    "dialogAction": {
        "slotElicitationStyle": "string",
        "slotToElicit": "string",
        "subSlotToElicit": {
            "name": "string",
            "subSlotToElicit": "ElicitSubSlot"
        },
        "type": "string"
    },
    "intent": {
        "confirmationState": "string",
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "subSlots": {
                    "string" : "Slot"
                },
                "value": {
                    "interpretedValue": "string",
                    "originalValue": "string",
                    "resolvedValues": [ "string" ]
                },
                "values": [
                    "Slot"
                ]
            }
        },
        "state": "string"
    },
    "originatingRequestId": "string",
    "runtimeHints": {
        "slotHints": {
            "string" : {
                "string" : {
                    "runtimesHintValues": [
                        {
                            "phrase": "string"
                        }
                    ],
                    "subSlotHints": {
                        "string" : "RuntimeHintDetails"
                    }
                }
            }
        }
    },
    "sessionAttributes": {
        "string" : "string"
    }
},
"welcomeMessages": [
```

```
{  
    "content": "string",  
    "contentType": "string",  
    "imageResponseCard": {  
        "buttons": [  
            {  
                "text": "string",  
                "value": "string"  
            }  
        ],  
        "imageUrl": "string",  
        "subtitle": "string",  
        "title": "string"  
    }  
},  
]  
},  
"DisconnectionEvent": {  
    "clientTimestampMillis": number,  
    "eventId": "string"  
},  
"DTMFInputEvent": {  
    "clientTimestampMillis": number,  
    "eventId": "string",  
    "inputCharacter": "string"  
},  
"PlaybackCompletionEvent": {  
    "clientTimestampMillis": number,  
    "eventId": "string"  
},  
"TextInputEvent": {  
    "clientTimestampMillis": number,  
    "eventId": "string",  
    "text": "string"  
}  
}  
}
```

## URI Request Parameters

The request uses the following URI parameters.

### **botAliasId (p. 717)**

The alias identifier in use for the bot that processes the request.

Required: Yes

### **botId (p. 717)**

The identifier of the bot to process the request.

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### **conversationMode (p. 717)**

The conversation type that you are using the Amazon Lex V2. If the conversation mode is AUDIO you can send both audio and DTMF information. If the mode is TEXT you can only send text.

Valid Values: AUDIO | TEXT

### [localeId \(p. 717\)](#)

The locale where the session is in use.

Length Constraints: Minimum length of 1.

Required: Yes

### [sessionId \(p. 717\)](#)

The identifier of the user session that is having the conversation.

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: Yes

## Request Body

The request accepts the following data in JSON format.

### [requestEventStream \(p. 717\)](#)

Represents the stream of events to Amazon Lex V2 from your application. The events are encoded as HTTP/2 data frames.

Type: [StartConversationRequestEventStream \(p. 926\)](#) object

Required: Yes

## Response Syntax

```
HTTP/2 200
Content-type: application/json

{
  "responseEventStream": {
    "AccessDeniedException": {},
    "AudioResponseEvent": {
      "audioChunk": blob,
      "contentType": "string",
      "eventId": "string"
    },
    "BadGatewayException": {},
    "ConflictException": {},
    "DependencyFailedException": {},
    "HeartbeatEvent": {
      "eventId": "string"
    },
    "IntentResultEvent": {
      "eventId": "string",
      "inputMode": "string",
      "interpretations": [
        {
          "intent": {
            "confirmationState": "string",
            "name": "string",
            "value": "string"
          }
        }
      ]
    }
  }
}
```

```
"slots": {
    "string" : {
        "shape": "string",
        "subSlots": {
            "string" : "Slot"
        },
        "value": {
            "interpretedValue": "string",
            "originalValue": "string",
            "resolvedValues": [ "string" ]
        },
        "values": [
            "Slot"
        ]
    }
},
"state": "string"
},
"nluConfidence": {
    "score": number
},
"sentimentResponse": {
    "sentiment": "string",
    "sentimentScore": {
        "mixed": number,
        "negative": number,
        "neutral": number,
        "positive": number
    }
}
],
"requestAttributes": {
    "string" : "string"
},
"sessionId": "string",
"sessionState": {
    "activeContexts": [
        {
            "contextAttributes": {
                "string" : "string"
            },
            "name": "string",
            "timeToLive": {
                "timeToLiveInSeconds": number,
                "turnsToLive": number
            }
        }
    ],
    "dialogAction": {
        "slotElicitationStyle": "string",
        "slotToElicit": "string",
        "subSlotToElicit": {
            "name": "string",
            "subSlotToElicit": "ElicitSubSlot"
        },
        "type": "string"
    },
    "intent": {
        "confirmationState": "string",
        "name": "string",
        "slots": {
            "string" : {
                "shape": "string",
                "subSlots": {
                    "string" : "Slot"
                }
            }
        }
    }
}
```

```
        },
        "value": {
            "interpretedValue": "string",
            "originalValue": "string",
            "resolvedValues": [ "string" ]
        },
        "values": [
            "Slot"
        ]
    }
},
"state": "string"
},
"originatingRequestId": "string",
"runtimeHints": {
    "slotHints": {
        "string" : {
            "string" : {
                "runtimeHintValues": [
                    {
                        "phrase": "string"
                    }
                ],
                "subSlotHints": {
                    "string" : "RuntimeHintDetails"
                }
            }
        }
    }
},
"sessionAttributes": {
    "string" : "string"
}
}
},
"InternalServerException": {
},
"PlaybackInterruptionEvent": {
    "causedByEventId": "string",
    "eventId": "string",
    "eventReason": "string"
},
"ResourceNotFoundException": {
},
"TextResponseEvent": {
    "eventId": "string",
    "messages": [
        {
            "content": "string",
            "contentType": "string",
            "imageResponseCard": {
                "buttons": [
                    {
                        "text": "string",
                        "value": "string"
                    }
                ],
                "imageUrl": "string",
                "subtitle": "string",
                "title": "string"
            }
        }
    ]
},
"ThrottlingException": {
},
```

```
    "TranscriptEvent": {  
        "eventId": "string",  
        "transcript": "string"  
    },  
    "ValidationException": {  
    }  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### **responseEventStream (p. 720)**

Represents the stream of events from Amazon Lex V2 to your application. The events are encoded as HTTP/2 data frames.

Type: [StartConversationResponseEventStream \(p. 928\)](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 936\)](#).

### **AccessDeniedException**

HTTP Status Code: 403

### **InternalServerError**

HTTP Status Code: 500

### **ThrottlingException**

HTTP Status Code: 429

### **ValidationException**

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# Data Types

The following data types are supported by Amazon Lex Model Building V2:

- [AdvancedRecognitionSetting \(p. 732\)](#)
- [AggregatedUtterancesFilter \(p. 733\)](#)
- [AggregatedUtterancesSortBy \(p. 734\)](#)
- [AggregatedUtterancesSummary \(p. 735\)](#)
- [AllowedInputTypes \(p. 737\)](#)
- [AssociatedTranscript \(p. 738\)](#)
- [AssociatedTranscriptFilter \(p. 739\)](#)
- [AudioAndDTMFInputSpecification \(p. 740\)](#)
- [AudioLogDestination \(p. 741\)](#)
- [AudioLogSetting \(p. 742\)](#)
- [AudioSpecification \(p. 743\)](#)
- [BotAliasHistoryEvent \(p. 744\)](#)
- [BotAliasLocaleSettings \(p. 745\)](#)
- [BotAliasSummary \(p. 746\)](#)
- [BotExportSpecification \(p. 748\)](#)
- [BotFilter \(p. 749\)](#)
- [BotImportSpecification \(p. 750\)](#)
- [BotLocaleExportSpecification \(p. 752\)](#)
- [BotLocaleFilter \(p. 753\)](#)
- [BotLocaleHistoryEvent \(p. 754\)](#)
- [BotLocaleImportSpecification \(p. 755\)](#)
- [BotLocaleSortBy \(p. 757\)](#)
- [BotLocaleSummary \(p. 758\)](#)
- [BotRecommendationResults \(p. 760\)](#)
- [BotRecommendationResultStatistics \(p. 761\)](#)
- [BotRecommendationSummary \(p. 762\)](#)
- [BotSortBy \(p. 763\)](#)
- [BotSummary \(p. 764\)](#)
- [BotVersionLocaleDetails \(p. 766\)](#)
- [BotVersionSortBy \(p. 767\)](#)
- [BotVersionSummary \(p. 768\)](#)
- [BuiltInIntentSortBy \(p. 770\)](#)
- [BuiltInIntentSummary \(p. 771\)](#)
- [BuiltInSlotTypeSortBy \(p. 772\)](#)
- [BuiltInSlotTypeSummary \(p. 773\)](#)
- [Button \(p. 774\)](#)
- [CloudWatchLogGroupLogDestination \(p. 775\)](#)
- [CodeHookSpecification \(p. 776\)](#)
- [CompositeSlotTypeSetting \(p. 777\)](#)
- [Condition \(p. 778\)](#)
- [ConditionalBranch \(p. 779\)](#)
- [ConditionalSpecification \(p. 780\)](#)

- [ConversationLogSettings \(p. 781\)](#)
- [CustomPayload \(p. 782\)](#)
- [CustomVocabularyEntryId \(p. 783\)](#)
- [CustomVocabularyExportSpecification \(p. 784\)](#)
- [CustomVocabularyImportSpecification \(p. 785\)](#)
- [CustomVocabularyItem \(p. 786\)](#)
- [DataPrivacy \(p. 787\)](#)
- [DateRangeFilter \(p. 788\)](#)
- [DefaultConditionalBranch \(p. 789\)](#)
- [DialogAction \(p. 790\)](#)
- [DialogCodeHookInvocationSetting \(p. 791\)](#)
- [DialogCodeHookSettings \(p. 792\)](#)
- [DialogState \(p. 793\)](#)
- [DTMFSpecification \(p. 794\)](#)
- [ElicitationCodeHookInvocationSetting \(p. 795\)](#)
- [EncryptionSetting \(p. 796\)](#)
- [ExportFilter \(p. 797\)](#)
- [ExportResourceSpecification \(p. 798\)](#)
- [ExportSortBy \(p. 799\)](#)
- [ExportSummary \(p. 800\)](#)
- [ExternalSourceSetting \(p. 802\)](#)
- [FailedCustomVocabularyItem \(p. 803\)](#)
- [FulfillmentCodeHookSettings \(p. 804\)](#)
- [FulfillmentStartResponseSpecification \(p. 805\)](#)
- [FulfillmentUpdateResponseSpecification \(p. 806\)](#)
- [FulfillmentUpdatesSpecification \(p. 807\)](#)
- [GrammarSlotTypeSetting \(p. 808\)](#)
- [GrammarSlotTypeSource \(p. 809\)](#)
- [ImageResponseCard \(p. 810\)](#)
- [ImportFilter \(p. 812\)](#)
- [ImportResourceSpecification \(p. 813\)](#)
- [ImportSortBy \(p. 814\)](#)
- [ImportSummary \(p. 815\)](#)
- [InitialResponseSetting \(p. 817\)](#)
- [InputContext \(p. 818\)](#)
- [IntentClosingSetting \(p. 819\)](#)
- [IntentConfirmationSetting \(p. 820\)](#)
- [IntentFilter \(p. 823\)](#)
- [IntentOverride \(p. 824\)](#)
- [IntentSortBy \(p. 825\)](#)
- [IntentStatistics \(p. 826\)](#)
- [IntentSummary \(p. 827\)](#)
- [KendraConfiguration \(p. 829\)](#)
- [LambdaCodeHook \(p. 830\)](#)
- [LexTranscriptFilter \(p. 831\)](#)
- [Message \(p. 832\)](#)

- [MessageGroup \(p. 833\)](#)
- [MultipleValuesSetting \(p. 834\)](#)
- [NewCustomVocabularyItem \(p. 835\)](#)
- [ObfuscationSetting \(p. 836\)](#)
- [OutputContext \(p. 837\)](#)
- [PathFormat \(p. 838\)](#)
- [PlainTextMessage \(p. 839\)](#)
- [PostDialogCodeHookInvocationSpecification \(p. 840\)](#)
- [PostFulfillmentStatusSpecification \(p. 842\)](#)
- [Principal \(p. 844\)](#)
- [PromptAttemptSpecification \(p. 845\)](#)
- [PromptSpecification \(p. 846\)](#)
- [RecommendedIntentSummary \(p. 848\)](#)
- [RelativeAggregationDuration \(p. 849\)](#)
- [ResponseSpecification \(p. 850\)](#)
- [S3BucketLogDestination \(p. 851\)](#)
- [S3BucketTranscriptSource \(p. 852\)](#)
- [SampleUtterance \(p. 854\)](#)
- [SampleValue \(p. 855\)](#)
- [SentimentAnalysisSettings \(p. 856\)](#)
- [SlotCaptureSetting \(p. 857\)](#)
- [SlotDefaultValue \(p. 859\)](#)
- [SlotDefaultValueSpecification \(p. 860\)](#)
- [SlotFilter \(p. 861\)](#)
- [SlotPriority \(p. 862\)](#)
- [SlotSortBy \(p. 863\)](#)
- [SlotSummary \(p. 864\)](#)
- [SlotTypeFilter \(p. 866\)](#)
- [SlotTypeSortBy \(p. 867\)](#)
- [SlotTypeStatistics \(p. 868\)](#)
- [SlotTypeSummary \(p. 869\)](#)
- [SlotTypeValue \(p. 871\)](#)
- [SlotValue \(p. 872\)](#)
- [SlotValueElicitationSetting \(p. 873\)](#)
- [SlotValueOverride \(p. 875\)](#)
- [SlotValueRegexFilter \(p. 876\)](#)
- [SlotValueSelectionSetting \(p. 877\)](#)
- [Specifications \(p. 878\)](#)
- [SSMLMessage \(p. 879\)](#)
- [StillWaitingResponseSpecification \(p. 880\)](#)
- [SubSlotSetting \(p. 881\)](#)
- [SubSlotTypeComposition \(p. 882\)](#)
- [SubSlotValueElicitationSetting \(p. 883\)](#)
- [TextInputSpecification \(p. 884\)](#)
- [TextLogDestination \(p. 885\)](#)
- [TextLogSetting \(p. 886\)](#)

- [TranscriptFilter \(p. 887\)](#)
- [TranscriptSourceSetting \(p. 888\)](#)
- [UtteranceAggregationDuration \(p. 889\)](#)
- [VoiceSettings \(p. 890\)](#)
- [WaitAndContinueSpecification \(p. 891\)](#)

The following data types are supported by Amazon Lex Runtime V2:

- [ActiveContext \(p. 893\)](#)
- [ActiveContextTimeToLive \(p. 895\)](#)
- [AudioInputEvent \(p. 896\)](#)
- [AudioResponseEvent \(p. 898\)](#)
- [Button \(p. 899\)](#)
- [ConfidenceScore \(p. 900\)](#)
- [ConfigurationEvent \(p. 901\)](#)
- [DialogAction \(p. 903\)](#)
- [DisconnectionEvent \(p. 905\)](#)
- [DTMFInputEvent \(p. 906\)](#)
- [ElicitSubSlot \(p. 907\)](#)
- [HeartbeatEvent \(p. 908\)](#)
- [ImageResponseCard \(p. 909\)](#)
- [Intent \(p. 911\)](#)
- [IntentResultEvent \(p. 912\)](#)
- [Interpretation \(p. 914\)](#)
- [Message \(p. 915\)](#)
- [PlaybackCompletionEvent \(p. 916\)](#)
- [PlaybackInterruptionEvent \(p. 917\)](#)
- [RuntimeHintDetails \(p. 918\)](#)
- [RuntimeHints \(p. 919\)](#)
- [RuntimeHintValue \(p. 920\)](#)
- [SentimentResponse \(p. 921\)](#)
- [SentimentScore \(p. 922\)](#)
- [SessionState \(p. 923\)](#)
- [Slot \(p. 925\)](#)
- [StartConversationRequestEventStream \(p. 926\)](#)
- [StartConversationResponseEventStream \(p. 928\)](#)
- [TextInputEvent \(p. 931\)](#)
- [TextResponseEvent \(p. 932\)](#)
- [TranscriptEvent \(p. 933\)](#)
- [Value \(p. 934\)](#)

## Amazon Lex Model Building V2

The following data types are supported by Amazon Lex Model Building V2:

- [AdvancedRecognitionSetting \(p. 732\)](#)

- [AggregatedUtterancesFilter \(p. 733\)](#)
- [AggregatedUtterancesSortBy \(p. 734\)](#)
- [AggregatedUtterancesSummary \(p. 735\)](#)
- [AllowedInputTypes \(p. 737\)](#)
- [AssociatedTranscript \(p. 738\)](#)
- [AssociatedTranscriptFilter \(p. 739\)](#)
- [AudioAndDTMFIInputSpecification \(p. 740\)](#)
- [AudioLogDestination \(p. 741\)](#)
- [AudioLogSetting \(p. 742\)](#)
- [AudioSpecification \(p. 743\)](#)
- [BotAliasHistoryEvent \(p. 744\)](#)
- [BotAliasLocaleSettings \(p. 745\)](#)
- [BotAliasSummary \(p. 746\)](#)
- [BotExportSpecification \(p. 748\)](#)
- [BotFilter \(p. 749\)](#)
- [BotImportSpecification \(p. 750\)](#)
- [BotLocaleExportSpecification \(p. 752\)](#)
- [BotLocaleFilter \(p. 753\)](#)
- [BotLocaleHistoryEvent \(p. 754\)](#)
- [BotLocaleImportSpecification \(p. 755\)](#)
- [BotLocaleSortBy \(p. 757\)](#)
- [BotLocaleSummary \(p. 758\)](#)
- [BotRecommendationResults \(p. 760\)](#)
- [BotRecommendationResultStatistics \(p. 761\)](#)
- [BotRecommendationSummary \(p. 762\)](#)
- [BotSortBy \(p. 763\)](#)
- [BotSummary \(p. 764\)](#)
- [BotVersionLocaleDetails \(p. 766\)](#)
- [BotVersionSortBy \(p. 767\)](#)
- [BotVersionSummary \(p. 768\)](#)
- [BuiltInIntentSortBy \(p. 770\)](#)
- [BuiltInIntentSummary \(p. 771\)](#)
- [BuiltInSlotTypeSortBy \(p. 772\)](#)
- [BuiltInSlotTypeSummary \(p. 773\)](#)
- [Button \(p. 774\)](#)
- [CloudWatchLogGroupLogDestination \(p. 775\)](#)
- [CodeHookSpecification \(p. 776\)](#)
- [CompositeSlotTypeSetting \(p. 777\)](#)
- [Condition \(p. 778\)](#)
- [ConditionalBranch \(p. 779\)](#)
- [ConditionalSpecification \(p. 780\)](#)
- [ConversationLogSettings \(p. 781\)](#)
- [CustomPayload \(p. 782\)](#)
- [CustomVocabularyEntryId \(p. 783\)](#)
- [CustomVocabularyExportSpecification \(p. 784\)](#)
- [CustomVocabularyImportSpecification \(p. 785\)](#)

- [CustomVocabularyItem \(p. 786\)](#)
- [DataPrivacy \(p. 787\)](#)
- [DateRangeFilter \(p. 788\)](#)
- [DefaultConditionalBranch \(p. 789\)](#)
- [DialogAction \(p. 790\)](#)
- [DialogCodeHookInvocationSetting \(p. 791\)](#)
- [DialogCodeHookSettings \(p. 792\)](#)
- [DialogState \(p. 793\)](#)
- [DTMFSpecification \(p. 794\)](#)
- [ElicitationCodeHookInvocationSetting \(p. 795\)](#)
- [EncryptionSetting \(p. 796\)](#)
- [ExportFilter \(p. 797\)](#)
- [ExportResourceSpecification \(p. 798\)](#)
- [ExportSortBy \(p. 799\)](#)
- [ExportSummary \(p. 800\)](#)
- [ExternalSourceSetting \(p. 802\)](#)
- [FailedCustomVocabularyItem \(p. 803\)](#)
- [FulfillmentCodeHookSettings \(p. 804\)](#)
- [FulfillmentStartResponseSpecification \(p. 805\)](#)
- [FulfillmentUpdateResponseSpecification \(p. 806\)](#)
- [FulfillmentUpdatesSpecification \(p. 807\)](#)
- [GrammarSlotTypeSetting \(p. 808\)](#)
- [GrammarSlotTypeSource \(p. 809\)](#)
- [ImageResponseCard \(p. 810\)](#)
- [ImportFilter \(p. 812\)](#)
- [ImportResourceSpecification \(p. 813\)](#)
- [ImportSortBy \(p. 814\)](#)
- [ImportSummary \(p. 815\)](#)
- [InitialResponseSetting \(p. 817\)](#)
- [InputContext \(p. 818\)](#)
- [IntentClosingSetting \(p. 819\)](#)
- [IntentConfirmationSetting \(p. 820\)](#)
- [IntentFilter \(p. 823\)](#)
- [IntentOverride \(p. 824\)](#)
- [IntentSortBy \(p. 825\)](#)
- [IntentStatistics \(p. 826\)](#)
- [IntentSummary \(p. 827\)](#)
- [KendraConfiguration \(p. 829\)](#)
- [LambdaCodeHook \(p. 830\)](#)
- [LexTranscriptFilter \(p. 831\)](#)
- [Message \(p. 832\)](#)
- [MessageGroup \(p. 833\)](#)
- [MultipleValuesSetting \(p. 834\)](#)
- [NewCustomVocabularyItem \(p. 835\)](#)
- [ObfuscationSetting \(p. 836\)](#)
- [OutputContext \(p. 837\)](#)

- [PathFormat \(p. 838\)](#)
- [PlainTextMessage \(p. 839\)](#)
- [PostDialogCodeHookInvocationSpecification \(p. 840\)](#)
- [PostFulfillmentStatusSpecification \(p. 842\)](#)
- [Principal \(p. 844\)](#)
- [PromptAttemptSpecification \(p. 845\)](#)
- [PromptSpecification \(p. 846\)](#)
- [RecommendedIntentSummary \(p. 848\)](#)
- [RelativeAggregationDuration \(p. 849\)](#)
- [ResponseSpecification \(p. 850\)](#)
- [S3BucketLogDestination \(p. 851\)](#)
- [S3BucketTranscriptSource \(p. 852\)](#)
- [SampleUtterance \(p. 854\)](#)
- [SampleValue \(p. 855\)](#)
- [SentimentAnalysisSettings \(p. 856\)](#)
- [SlotCaptureSetting \(p. 857\)](#)
- [SlotDefaultValue \(p. 859\)](#)
- [SlotDefaultValueSpecification \(p. 860\)](#)
- [SlotFilter \(p. 861\)](#)
- [SlotPriority \(p. 862\)](#)
- [SlotSortBy \(p. 863\)](#)
- [SlotSummary \(p. 864\)](#)
- [SlotTypeFilter \(p. 866\)](#)
- [SlotTypeSortBy \(p. 867\)](#)
- [SlotTypeStatistics \(p. 868\)](#)
- [SlotTypeSummary \(p. 869\)](#)
- [SlotTypeValue \(p. 871\)](#)
- [SlotValue \(p. 872\)](#)
- [SlotValueElicitationSetting \(p. 873\)](#)
- [SlotValueOverride \(p. 875\)](#)
- [SlotValueRegexFilter \(p. 876\)](#)
- [SlotValueSelectionSetting \(p. 877\)](#)
- [Specifications \(p. 878\)](#)
- [SSMLMessage \(p. 879\)](#)
- [StillWaitingResponseSpecification \(p. 880\)](#)
- [SubSlotSetting \(p. 881\)](#)
- [SubSlotTypeComposition \(p. 882\)](#)
- [SubSlotValueElicitationSetting \(p. 883\)](#)
- [TextInputSpecification \(p. 884\)](#)
- [TextLogDestination \(p. 885\)](#)
- [TextLogSetting \(p. 886\)](#)
- [TranscriptFilter \(p. 887\)](#)
- [TranscriptSourceSetting \(p. 888\)](#)
- [UtteranceAggregationDuration \(p. 889\)](#)
- [VoiceSettings \(p. 890\)](#)
- [WaitAndContinueSpecification \(p. 891\)](#)



## AdvancedRecognitionSetting

Service: Amazon Lex Model Building V2

Provides settings that enable advanced recognition settings for slot values.

### Contents

#### **audioRecognitionStrategy**

Enables using the slot values as a custom vocabulary for recognizing user utterances.

Type: String

Valid Values: UseSlotValuesAsCustomVocabulary

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AggregatedUtterancesFilter

Service: Amazon Lex Model Building V2

Filters responses returned by the ListAggregatedUtterances operation.

### Contents

#### **name**

The name of the field to filter the utterance list.

Type: String

Valid Values: Utterance

Required: Yes

#### **operator**

The operator to use for the filter. Specify EQ when the ListAggregatedUtterances operation should return only utterances that equal the specified value. Specify CO when the ListAggregatedUtterances operation should return utterances that contain the specified value.

Type: String

Valid Values: CO | EQ

Required: Yes

#### **values**

The value to use for filtering the list of bots.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[0-9a-zA-Z\_()\\s-]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AggregatedUtterancesSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of utterances.

### Contents

#### attribute

The utterance attribute to sort by.

Type: String

Valid Values: HitCount | MissedCount

Required: Yes

#### order

Specifies whether to sort the aggregated utterances in ascending or descending order.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AggregatedUtterancesSummary

Service: Amazon Lex Model Building V2

Provides summary information for aggregated utterances. The `ListAggregatedUtterances` operations combines all instances of the same utterance into a single aggregated summary.

### Contents

#### **containsDataFromDeletedResources**

Aggregated utterance data may contain utterances from versions of your bot that have since been deleted. When the aggregated contains this kind of data, this field is set to true.

Type: Boolean

Required: No

#### **hitCount**

The number of times that the utterance was detected by Amazon Lex during the time period. When an utterance is detected, it activates an intent or a slot.

Type: Integer

Required: No

#### **missedCount**

The number of times that the utterance was missed by Amazon Lex. An utterance is missed when it doesn't activate an intent or slot.

Type: Integer

Required: No

#### **utterance**

The text of the utterance. If the utterance was used with the `RecognizeUtterance` operation, the text is the transcription of the audio utterance.

Type: String

Required: No

#### **utteranceFirstRecordedInAggregationDuration**

The date and time that the utterance was first recorded in the time window for aggregation. An utterance may have been sent to Amazon Lex before that time, but only utterances within the time window are counted.

Type: Timestamp

Required: No

#### **utteranceLastRecordedInAggregationDuration**

The last date and time that an utterance was recorded in the time window for aggregation. An utterance may be sent to Amazon Lex after that time, but only utterances within the time window are counted.

Type: Timestamp

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AllowedInputTypes

Service: Amazon Lex Model Building V2

Specifies the allowed input types.

### Contents

#### allowAudioInput

Indicates whether audio input is allowed.

Type: Boolean

Required: Yes

#### allowDTMFInput

Indicates whether DTMF input is allowed.

Type: Boolean

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AssociatedTranscript

Service: Amazon Lex Model Building V2

The object containing information that associates the recommended intent/slot type with a conversation.

### Contents

#### **transcript**

The content of the transcript that meets the search filter criteria. For the JSON format of the transcript, see [Output transcript format](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 6000000.

Pattern: .\*

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AssociatedTranscriptFilter

Service: Amazon Lex Model Building V2

Filters to search for the associated transcript.

### Contents

#### **name**

The name of the field to use for filtering. The allowed names are IntentId and SlotTypeId.

Type: String

Valid Values: IntentId | SlotTypeId

Required: Yes

#### **values**

The values to use to filter the transcript.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[0-9a-zA-Z\_()\\s-]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AudioAndDTMFInputSpecification

Service: Amazon Lex Model Building V2

Specifies the audio and DTMF input specification.

### Contents

#### **audioSpecification**

Specifies the settings on audio input.

Type: [AudioSpecification \(p. 743\)](#) object

Required: No

#### **dtmfSpecification**

Specifies the settings on DTMF input.

Type: [DTMFSpecification \(p. 794\)](#) object

Required: No

#### **startTimeoutMs**

Time for which a bot waits before assuming that the customer isn't going to speak or press a key.  
This timeout is shared between Audio and DTMF inputs.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AudioLogDestination

Service: Amazon Lex Model Building V2

The location of audio log files collected when conversation logging is enabled for a bot.

### Contents

#### s3Bucket

The Amazon S3 bucket where the audio log files are stored. The IAM role specified in the `roleArn` parameter of the [CreateBot](#) operation must have permission to write to this bucket.

Type: [S3BucketLogDestination \(p. 851\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AudioLogSetting

Service: Amazon Lex Model Building V2

Settings for logging audio of conversations between Amazon Lex and a user. You specify whether to log audio and the Amazon S3 bucket where the audio file is stored.

### Contents

#### **destination**

The location of audio log files collected when conversation logging is enabled for a bot.

Type: [AudioLogDestination \(p. 741\)](#) object

Required: Yes

#### **enabled**

Determines whether audio logging is enabled for the bot.

Type: Boolean

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AudioSpecification

Service: Amazon Lex Model Building V2

Specifies the audio input specifications.

### Contents

#### **endTimeTimeoutMs**

Time for which a bot waits after the customer stops speaking to assume the utterance is finished.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

#### **maxLengthMs**

Time for how long Amazon Lex waits before speech input is truncated and the speech is returned to application.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotAliasHistoryEvent

Service: Amazon Lex Model Building V2

Provides a record of an event that affects a bot alias. For example, when the version of a bot that the alias points to changes.

### Contents

#### **botVersion**

The version of the bot that was used in the event.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: No

#### **endDate**

The date and time that the event ended.

Type: Timestamp

Required: No

#### **startDate**

The date and time that the event started.

Type: Timestamp

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotAliasLocaleSettings

Service: Amazon Lex Model Building V2

Specifies settings that are unique to a locale. For example, you can use different Lambda function depending on the bot's locale.

### Contents

#### **codeHookSpecification**

Specifies the Lambda function that should be used in the locale.

Type: [CodeHookSpecification \(p. 776\)](#) object

Required: No

#### **enabled**

Determines whether the locale is enabled for the bot. If the value is `false`, the locale isn't available for use.

Type: Boolean

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotAliasSummary

Service: Amazon Lex Model Building V2

Summary information about bot aliases returned from the [ListBotAliases](#) operation.

### Contents

#### **botAliasId**

The unique identifier assigned to the bot alias. You can use this ID to get detailed information about the alias using the [DescribeBotAlias](#) operation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^(\bTSTALIASID\b|[0-9a-zA-Z]+)\$

Required: No

#### **botAliasName**

The name of the bot alias.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### **botAliasStatus**

The current state of the bot alias. If the status is Available, the alias is ready for use.

Type: String

Valid Values: Creating | Available | Deleting | Failed

Required: No

#### **botVersion**

The version of the bot that the bot alias references.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: No

#### **creationDateTime**

A timestamp of the date and time that the bot alias was created.

Type: Timestamp

Required: No

#### **description**

The description of the bot alias.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

**lastUpdatedDateTime**

A timestamp of the date and time that the bot alias was last updated.

Type: Timestamp

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotExportSpecification

Service: Amazon Lex Model Building V2

Provides the identity of a the bot that was exported.

### Contents

#### botId

The identifier of the bot assigned by Amazon Lex.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### botVersion

The version of the bot that was exported. This will be either DRAFT or the version number.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotFilter

Service: Amazon Lex Model Building V2

Filters the responses returned by the `ListBots` operation.

### Contents

#### **name**

The name of the field to filter the list of bots.

Type: String

Valid Values: BotName

Required: Yes

#### **operator**

The operator to use for the filter. Specify EQ when the `ListBots` operation should return only aliases that equal the specified value. Specify CO when the `ListBots` operation should return aliases that contain the specified value.

Type: String

Valid Values: CO | EQ

Required: Yes

#### **values**

The value to use for filtering the list of bots.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[0-9a-zA-Z\_()\\s-]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotImportSpecification

Service: Amazon Lex Model Building V2

Provides the bot parameters required for importing a bot.

### Contents

#### botName

The name that Amazon Lex should use for the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### botTags

A list of tags to add to the bot. You can only add tags when you import a bot. You can't use the UpdateBot operation to update tags. To update tags, use the TagResource operation.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

#### dataPrivacy

By default, data stored by Amazon Lex is encrypted. The DataPrivacy structure provides settings that determine how Amazon Lex handles special cases of securing the data for your bot.

Type: [DataPrivacy \(p. 787\)](#) object

Required: Yes

#### idleSessionTTLInSeconds

The time, in seconds, that Amazon Lex should keep information about a user's conversation with the bot.

A user interaction remains active for the amount of time specified. If no conversation occurs during this time, the session expires and Amazon Lex deletes any data provided before the timeout.

You can specify between 60 (1 minute) and 86,400 (24 hours) seconds.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 86400.

Required: No

#### roleArn

The Amazon Resource Name (ARN) of the IAM role used to build and run the bot.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:iam::[0-9]{12}:role/.\*\$

Required: Yes

#### **testBotAliasTags**

A list of tags to add to the test alias for a bot. You can only add tags when you import a bot. You can't use the UpdateAlias operation to update tags. To update tags on the test alias, use the TagResource operation.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotLocaleExportSpecification

Service: Amazon Lex Model Building V2

Provides the bot locale parameters required for exporting a bot locale.

### Contents

#### botId

The identifier of the bot to create the locale for.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### botVersion

The version of the bot to export.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### localeId

The identifier of the language and locale to export. The string must match one of the locales in the bot.

Type: String

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotLocaleFilter

Service: Amazon Lex Model Building V2

Filters responses returned by the `ListBotLocales` operation.

### Contents

#### **name**

The name of the field to filter the list of bots.

Type: String

Valid Values: `BotLocaleName`

Required: Yes

#### **operator**

The operator to use for the filter. Specify `EQ` when the `ListBotLocales` operation should return only aliases that equal the specified value. Specify `CO` when the `ListBotLocales` operation should return aliases that contain the specified value.

Type: String

Valid Values: `CO` | `EQ`

Required: Yes

#### **values**

The value to use for filtering the list of bots.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\0-9a-zA-Z_()\\s-]+$`

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotLocaleHistoryEvent

Service: Amazon Lex Model Building V2

Provides information about an event that occurred affecting the bot locale.

### Contents

#### **event**

A description of the event that occurred.

Type: String

Required: Yes

#### **eventDate**

A timestamp of the date and time that the event occurred.

Type: Timestamp

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotLocaleImportSpecification

Service: Amazon Lex Model Building V2

Provides the bot locale parameters required for importing a bot locale.

### Contents

#### **botId**

The identifier of the bot to import the locale to.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botVersion**

The version of the bot to import the locale to. This can only be the DRAFT version of the bot.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### **localeId**

The identifier of the language and locale that the bot will be used in. The string must match one of the supported locales. All of the intents, slot types, and slots used in the bot must have the same locale. For more information, see [Supported languages](#).

Type: String

Required: Yes

#### **nluIntentConfidenceThreshold**

Determines the threshold where Amazon Lex will insert the AMAZON.FallbackIntent, AMAZON.KendraSearchIntent, or both when returning alternative intents.

AMAZON.FallbackIntent and AMAZON.KendraSearchIntent are only inserted if they are configured for the bot.

For example, suppose a bot is configured with the confidence threshold of 0.80 and the AMAZON.FallbackIntent. Amazon Lex returns three alternative intents with the following confidence scores: IntentA (0.70), IntentB (0.60), IntentC (0.50). The response from the PostText operation would be:

- AMAZON.FallbackIntent
- IntentA
- IntentB
- IntentC

Type: Double

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

**voiceSettings**

Defines settings for using an Amazon Polly voice to communicate with a user.

Type: [VoiceSettings \(p. 890\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotLocaleSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of bot locales.

### Contents

#### attribute

The bot locale attribute to sort by.

Type: String

Valid Values: BotLocaleName

Required: Yes

#### order

Specifies whether to sort the bot locales in ascending or descending order.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotLocaleSummary

Service: Amazon Lex Model Building V2

Summary information about bot locales returned by the [ListBotLocales](#) operation.

### Contents

#### **botLocaleStatus**

The current status of the bot locale. When the status is Built the locale is ready for use.

Type: String

Valid Values: Creating | Building | Built | ReadyExpressTesting | Failed | Deleting | NotBuilt | Importing | Processing

Required: No

#### **description**

The description of the bot locale.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **lastBuildSubmittedDateTime**

A timestamp of the date and time that the bot locale was last built.

Type: Timestamp

Required: No

#### **lastUpdatedDateTime**

A timestamp of the date and time that the bot locale was last updated.

Type: Timestamp

Required: No

#### **localeId**

The language and locale of the bot locale.

Type: String

Required: No

#### **localeName**

The name of the bot locale.

Type: String

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotRecommendationResults

Service: Amazon Lex Model Building V2

The object representing the URL of the bot definition, the URL of the associated transcript, and a statistical summary of the bot recommendation results.

### Contents

#### **associatedTranscriptsUrl**

The presigned url link of the associated transcript.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

#### **botLocaleExportUrl**

The presigned URL link of the recommended bot definition.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

#### **statistics**

The statistical summary of the bot recommendation results.

Type: [BotRecommendationResultStatistics \(p. 761\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotRecommendationResultStatistics

Service: Amazon Lex Model Building V2

A statistical summary of the bot recommendation results.

### Contents

#### intents

Statistical information about the intents associated with the bot recommendation results.

Type: [IntentStatistics \(p. 826\)](#) object

Required: No

#### slotTypes

Statistical information about the slot types associated with the bot recommendation results.

Type: [SlotTypeStatistics \(p. 868\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotRecommendationSummary

Service: Amazon Lex Model Building V2

A summary of the bot recommendation.

### Contents

#### **botRecommendationId**

The unique identifier of the bot recommendation to be updated.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### **botRecommendationStatus**

The status of the bot recommendation.

If the status is Failed, then the reasons for the failure are listed in the failureReasons field.

Type: String

Valid Values: Processing | Deleting | Deleted | Downloading | Updating | Available | Failed | Stopping | Stopped

Required: Yes

#### **creationDateTime**

A timestamp of the date and time that the bot recommendation was created.

Type: Timestamp

Required: No

#### **lastUpdatedDateTime**

A timestamp of the date and time that the bot recommendation was last updated.

Type: Timestamp

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of bots.

### Contents

#### attribute

The attribute to use to sort the list of bots.

Type: String

Valid Values: BotName

Required: Yes

#### order

The order to sort the list. You can choose ascending or descending.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotSummary

Service: Amazon Lex Model Building V2

Summary information about a bot returned by the [ListBots](#) operation.

### Contents

#### **botId**

The unique identifier assigned to the bot. Use this ID to get detailed information about the bot with the [DescribeBot](#) operation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **botName**

The name of the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### **botStatus**

The current status of the bot. When the status is Available the bot is ready for use.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

Required: No

#### **description**

The description of the bot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **lastUpdatedDateTime**

The date and time that the bot was last updated.

Type: Timestamp

Required: No

#### **latestBotVersion**

The latest numerical version in use for the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^[0-9]+\$

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotVersionLocaleDetails

Service: Amazon Lex Model Building V2

The version of a bot used for a bot locale.

### Contents

#### sourceBotVersion

The version of a bot used for a bot locale.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotVersionSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of bot versions.

### Contents

#### attribute

The attribute to use to sort the list of versions.

Type: String

Valid Values: BotVersion

Required: Yes

#### order

The order to sort the list. You can specify ascending or descending order.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BotVersionSummary

Service: Amazon Lex Model Building V2

Summary information about a bot version returned by the [ListBotVersions](#) operation.

### Contents

#### botName

The name of the bot associated with the version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### botStatus

The status of the bot. When the status is available, the version of the bot is ready for use.

Type: String

Valid Values: Creating | Available | Inactive | Deleting | Failed | Versioning | Importing

Required: No

#### botVersion

The numeric version of the bot, or DRAFT to indicate that this is the version of the bot that can be updated..

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: No

#### creationDateTime

A timestamp of the date and time that the version was created.

Type: Timestamp

Required: No

#### description

The description of the version.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BuiltInIntentSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of built-in intents.

### Contents

#### attribute

The attribute to use to sort the list of built-in intents.

Type: String

Valid Values: IntentSignature

Required: Yes

#### order

The order to sort the list. You can specify ascending or descending order.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BuiltInIntentSummary

Service: Amazon Lex Model Building V2

Provides summary information about a built-in intent for the [ListBuiltInIntents](#) operation.

### Contents

#### **description**

The description of the intent.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **intentSignature**

The signature of the built-in intent. Use this to specify the parent intent of a derived intent.

Type: String

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BuiltInSlotTypeSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of built-in slot types.

### Contents

#### attribute

The attribute to use to sort the list of built-in intents.

Type: String

Valid Values: SlotTypeSignature

Required: Yes

#### order

The order to sort the list. You can choose ascending or descending.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BuiltInSlotTypeSummary

Service: Amazon Lex Model Building V2

Provides summary information about a built-in slot type for the [ListBuiltInSlotTypes](#) operation.

### Contents

#### **description**

The description of the built-in slot type.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **slotTypeSignature**

The signature of the built-in slot type. Use this to specify the parent slot type of a derived slot type.

Type: String

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Button

Service: Amazon Lex Model Building V2

Describes a button to use on a response card used to gather slot values from a user.

### Contents

#### **text**

The text that appears on the button. Use this to tell the user what value is returned when they choose this button.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 50.

Required: Yes

#### **value**

The value returned to Amazon Lex when the user chooses this button. This must be one of the slot values configured for the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 50.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CloudWatchLogGroupLogDestination

Service: Amazon Lex Model Building V2

The Amazon CloudWatch Logs log group where the text and metadata logs are delivered. The log group must exist before you enable logging.

### Contents

#### cloudWatchLogGroupArn

The Amazon Resource Name (ARN) of the log group where text and metadata logs are delivered.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^arn:[\w\[-]+:logs:[\w\[-]+:[\d]{12}:log-group:[\.\\_\-\#/A-Za-z0-9]{1,512}(?:\:|^\*)?\$\$

Required: Yes

#### logPrefix

The prefix of the log stream name within the log group that you specified

Type: String

Length Constraints: Maximum length of 1024.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CodeHookSpecification

Service: Amazon Lex Model Building V2

Contains information about code hooks that Amazon Lex calls during a conversation.

### Contents

#### **lambdaCodeHook**

Specifies a Lambda function that verifies requests to a bot or fulfills the user's request to a bot.

Type: [LambdaCodeHook \(p. 830\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CompositeSlotTypeSetting

Service: Amazon Lex Model Building V2

A composite slot is a combination of two or more slots that capture multiple pieces of information in a single user input.

### Contents

#### subSlots

Subslots in the composite slot.

Type: Array of [SubSlotTypeComposition \(p. 882\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 6 items.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Condition

Service: Amazon Lex Model Building V2

Provides an expression that evaluates to true or false.

### Contents

#### **expressionString**

The expression string that is evaluated.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ConditionalBranch

Service: Amazon Lex Model Building V2

A set of actions that Amazon Lex should run if the condition is matched.

### Contents

#### condition

Contains the expression to evaluate. If the condition is true, the branch's actions are taken.

Type: [Condition \(p. 778\)](#) object

Required: Yes

#### name

The name of the branch.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### nextStep

The next step in the conversation.

Type: [DialogState \(p. 793\)](#) object

Required: Yes

#### response

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ConditionalSpecification

Service: Amazon Lex Model Building V2

Provides a list of conditional branches. Branches are evaluated in the order that they are entered in the list. The first branch with a condition that evaluates to true is executed. The last branch in the list is the default branch. The default branch should not have any condition expression. The default branch is executed if no other branch has a matching condition.

### Contents

#### active

Determines whether a conditional branch is active. When active is false, the conditions are not evaluated.

Type: Boolean

Required: Yes

#### conditionalBranches

A list of conditional branches. A conditional branch is made up of a condition, a response and a next step. The response and next step are executed when the condition is true.

Type: Array of [ConditionalBranch \(p. 779\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Required: Yes

#### defaultBranch

The conditional branch that should be followed when the conditions for other branches are not satisfied. A conditional branch is made up of a condition, a response and a next step.

Type: [DefaultConditionalBranch \(p. 789\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ConversationLogSettings

Service: Amazon Lex Model Building V2

Configures conversation logging that saves audio, text, and metadata for the conversations with your users.

### Contents

#### **audioLogSettings**

The Amazon S3 settings for logging audio to an S3 bucket.

Type: Array of [AudioLogSetting \(p. 742\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

#### **textLogSettings**

The Amazon CloudWatch Logs settings for logging text and metadata.

Type: Array of [TextLogSetting \(p. 886\)](#) objects

Array Members: Fixed number of 1 item.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CustomPayload

Service: Amazon Lex Model Building V2

A custom response string that Amazon Lex sends to your application. You define the content and structure the string.

### Contents

#### **value**

The string that is sent to your application.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CustomVocabularyEntryId

Service: Amazon Lex Model Building V2

The unique entry identifier for the custom vocabulary items.

### Contents

#### itemId

The unique item identifier for the custom vocabulary items.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CustomVocabularyExportSpecification

Service: Amazon Lex Model Building V2

Provides the parameters required for exporting a custom vocabulary.

### Contents

#### botId

The identifier of the bot that contains the custom vocabulary to export.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### botVersion

The version of the bot that contains the custom vocabulary to export.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Pattern: ^(DRAFT|[0-9]+)\$

Required: Yes

#### localeId

The locale of the bot that contains the custom vocabulary to export.

Type: String

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CustomVocabularyImportSpecification

Service: Amazon Lex Model Building V2

Provides the parameters required for importing a custom vocabulary.

### Contents

#### botId

The identifier of the bot to import the custom vocabulary to.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

#### botVersion

The version of the bot to import the custom vocabulary to.

Type: String

Length Constraints: Fixed length of 5.

Pattern: ^DRAFT\$

Required: Yes

#### localeId

The identifier of the locale to import the custom vocabulary to. The value must be en\_GB.

Type: String

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## CustomVocabularyItem

Service: Amazon Lex Model Building V2

The unique custom vocabulary item from the custom vocabulary list.

### Contents

#### displayAs

The DisplayAs value for the custom vocabulary item from the custom vocabulary list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: No

#### itemId

The unique item identifier for the custom vocabulary item from the custom vocabulary list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Required: Yes

#### phrase

The unique phrase for the custom vocabulary item from the custom vocabulary list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: Yes

#### weight

The weight assigned for the custom vocabulary item from the custom vocabulary list.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 3.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DataPrivacy

Service: Amazon Lex Model Building V2

By default, data stored by Amazon Lex is encrypted. The DataPrivacy structure provides settings that determine how Amazon Lex handles special cases of securing the data for your bot.

### Contents

#### childDirected

For each Amazon Lex bot created with the Amazon Lex Model Building Service, you must specify whether your use of Amazon Lex is related to a website, program, or other application that is directed or targeted, in whole or in part, to children under age 13 and subject to the Children's Online Privacy Protection Act (COPPA) by specifying `true` or `false` in the `childDirected` field. By specifying `true` in the `childDirected` field, you confirm that your use of Amazon Lex **is** related to a website, program, or other application that is directed or targeted, in whole or in part, to children under age 13 and subject to COPPA. By specifying `false` in the `childDirected` field, you confirm that your use of Amazon Lex **is not** related to a website, program, or other application that is directed or targeted, in whole or in part, to children under age 13 and subject to COPPA. You may not specify a default value for the `childDirected` field that does not accurately reflect whether your use of Amazon Lex is related to a website, program, or other application that is directed or targeted, in whole or in part, to children under age 13 and subject to COPPA. If your use of Amazon Lex relates to a website, program, or other application that is directed in whole or in part, to children under age 13, you must obtain any required verifiable parental consent under COPPA. For information regarding the use of Amazon Lex in connection with websites, programs, or other applications that are directed or targeted, in whole or in part, to children under age 13, see the [Amazon Lex FAQ](#).

Type: Boolean

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DateRangeFilter

Service: Amazon Lex Model Building V2

The object used for specifying the data range that the customer wants Amazon Lex to read through in the input transcripts.

### Contents

#### **endDateTime**

A timestamp indicating the end date for the date range filter.

Type: Timestamp

Required: Yes

#### **startDateTime**

A timestamp indicating the start date for the date range filter.

Type: Timestamp

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DefaultConditionalBranch

Service: Amazon Lex Model Building V2

A set of actions that Amazon Lex should run if none of the other conditions are met.

### Contents

#### nextStep

The next step in the conversation.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### response

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DialogAction

Service: Amazon Lex Model Building V2

Defines the action that the bot executes at runtime when the conversation reaches this step.

### Contents

#### slotToElicit

If the dialog action is ElicitSlot, defines the slot to elicit from the user.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### suppressNextMessage

When true the next message for the intent is not used.

Type: Boolean

Required: No

#### type

The action that the bot should execute.

Type: String

Valid Values: ElicitIntent | StartIntent | ElicitSlot | EvaluateConditional | InvokeDialogCodeHook | ConfirmIntent | FulfillIntent | CloseIntent | EndConversation

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DialogCodeHookInvocationSetting

Service: Amazon Lex Model Building V2

Settings that specify the dialog code hook that is called by Amazon Lex at a step of the conversation.

### Contents

#### **active**

Determines whether a dialog code hook is used when the intent is activated.

Type: Boolean

Required: Yes

#### **enableCodeHookInvocation**

Indicates whether a Lambda function should be invoked for the dialog.

Type: Boolean

Required: Yes

#### **invocationLabel**

A label that indicates the dialog step from which the dialog code hook is happening.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### **postCodeHookSpecification**

Contains the responses and actions that Amazon Lex takes after the Lambda function is complete.

Type: [PostDialogCodeHookInvocationSpecification \(p. 840\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DialogCodeHookSettings

Service: Amazon Lex Model Building V2

Settings that determine the Lambda function that Amazon Lex uses for processing user responses.

### Contents

#### enabled

Enables the dialog code hook so that it processes user requests.

Type: Boolean

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DialogState

Service: Amazon Lex Model Building V2

The current state of the conversation with the user.

### Contents

#### **dialogAction**

Defines the action that the bot executes at runtime when the conversation reaches this step.

Type: [DialogAction \(p. 790\)](#) object

Required: No

#### **intent**

Override settings to configure the intent state.

Type: [IntentOverride \(p. 824\)](#) object

Required: No

#### **sessionAttributes**

Map of key/value pairs representing session-specific context information. It contains application information passed between Amazon Lex and a client application.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DTMF Specification

Service: Amazon Lex Model Building V2

Specifies the DTMF input specifications.

### Contents

#### **deletionCharacter**

The DTMF character that clears the accumulated DTMF digits and immediately ends the input.

Type: String

Pattern: ^[A-D0-9#\*]{1}\$

Required: Yes

#### **endCharacter**

The DTMF character that immediately ends input. If the user does not press this character, the input ends after the end timeout.

Type: String

Pattern: ^[A-D0-9#\*]{1}\$

Required: Yes

#### **endTimeoutMs**

How long the bot should wait after the last DTMF character input before assuming that the input has concluded.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

#### **maxLength**

The maximum number of DTMF digits allowed in an utterance.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1024.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ElicitationCodeHookInvocationSetting

Service: Amazon Lex Model Building V2

Settings that specify the dialog code hook that is called by Amazon Lex between eliciting slot values.

### Contents

#### enableCodeHookInvocation

Indicates whether a Lambda function should be invoked for the dialog.

Type: Boolean

Required: Yes

#### invocationLabel

A label that indicates the dialog step from which the dialog code hook is happening.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## EncryptionSetting

Service: Amazon Lex Model Building V2

The object representing the passwords that were used to encrypt the data related to the bot recommendation, as well as the KMS key ARN used to encrypt the associated metadata.

### Contents

#### **associatedTranscriptsPassword**

The password used to encrypt the associated transcript file.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

#### **botLocaleExportPassword**

The password used to encrypt the recommended bot recommendation file.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

#### **kmsKeyArn**

The KMS key ARN used to encrypt the metadata associated with the bot recommendation.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: ^arn:[\w\ -]+:kms:[\w\ -]+:[\d]{12}:(?:key\/[\w\ -]+\|alias\/[a-zA-Z0-9:\ \\_\-\ ]{1,256})\$

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExportFilter

Service: Amazon Lex Model Building V2

Filters the response from the [ListExports](#) operation

### Contents

#### name

The name of the field to use for filtering.

Type: String

Valid Values: ExportResourceType

Required: Yes

#### operator

The operator to use for the filter. Specify EQ when the [ListExports](#) operation should return only resource types that equal the specified value. Specify CO when the [ListExports](#) operation should return resource types that contain the specified value.

Type: String

Valid Values: CO | EQ

Required: Yes

#### values

The values to use to filter the response. The values must be Bot, BotLocale, or CustomVocabulary.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[0-9a-zA-Z\_()\\s-]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExportResourceSpecification

Service: Amazon Lex Model Building V2

Provides information about the bot or bot locale that you want to export. You can specify the `botExportSpecification` or the `botLocaleExportSpecification`, but not both.

### Contents

#### **botExportSpecification**

Parameters for exporting a bot.

Type: [BotExportSpecification \(p. 748\)](#) object

Required: No

#### **botLocaleExportSpecification**

Parameters for exporting a bot locale.

Type: [BotLocaleExportSpecification \(p. 752\)](#) object

Required: No

#### **customVocabularyExportSpecification**

The parameters required to export a custom vocabulary.

Type: [CustomVocabularyExportSpecification \(p. 784\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExportSortBy

Service: Amazon Lex Model Building V2

Provides information about sorting a list of exports.

### Contents

#### attribute

The export field to use for sorting.

Type: String

Valid Values: LastUpdatedDateTime

Required: Yes

#### order

The order to sort the list.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExportSummary

Service: Amazon Lex Model Building V2

Provides summary information about an export in an export list.

### Contents

#### **creationDateTime**

The date and time that the export was created.

Type: Timestamp

Required: No

#### **exportId**

The unique identifier that Amazon Lex assigned to the export.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **exportStatus**

The status of the export. When the status is Completed the export is ready to download.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

Required: No

#### **fileFormat**

The file format used in the export files.

Type: String

Valid Values: LexJson | TSV

Required: No

#### **lastUpdatedDateTime**

The date and time that the export was last updated.

Type: Timestamp

Required: No

#### **resourceSpecification**

Information about the bot or bot locale that was exported.

Type: [ExportResourceSpecification \(p. 798\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExternalSourceSetting

Service: Amazon Lex Model Building V2

Provides information about the external source of the slot type's definition.

### Contents

#### grammarSlotTypeSetting

Settings required for a slot type based on a grammar that you provide.

Type: [GrammarSlotTypeSetting \(p. 808\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FailedCustomVocabularyItem

Service: Amazon Lex Model Building V2

The unique failed custom vocabulary item from the custom vocabulary list.

### Contents

#### **errorCode**

The unique error code for the failed custom vocabulary item from the custom vocabulary list.

Type: String

Valid Values: DUPLICATE\_INPUT | RESOURCE\_DOES\_NOT\_EXIST |  
RESOURCE\_ALREADY\_EXISTS | INTERNAL\_SERVER\_FAILURE

Required: No

#### **errorMessage**

The error message for the failed custom vocabulary item from the custom vocabulary list.

Type: String

Required: No

#### **itemId**

The unique item identifier for the failed custom vocabulary item from the custom vocabulary list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FulfillmentCodeHookSettings

Service: Amazon Lex Model Building V2

Determines if a Lambda function should be invoked for a specific intent.

### Contents

#### active

Determines whether the fulfillment code hook is used. When active is false, the code hook doesn't run.

Type: Boolean

Required: No

#### enabled

Indicates whether a Lambda function should be invoked to fulfill a specific intent.

Type: Boolean

Required: Yes

#### fulfillmentUpdatesSpecification

Provides settings for update messages sent to the user for long-running Lambda fulfillment functions. Fulfillment updates can be used only with streaming conversations.

Type: [FulfillmentUpdatesSpecification \(p. 807\)](#) object

Required: No

#### postFulfillmentStatusSpecification

Provides settings for messages sent to the user after the Lambda fulfillment function completes. Post-fulfillment messages can be sent for both streaming and non-streaming conversations.

Type: [PostFulfillmentStatusSpecification \(p. 842\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FulfillmentStartResponseSpecification

Service: Amazon Lex Model Building V2

Provides settings for a message that is sent to the user when a fulfillment Lambda function starts running.

### Contents

#### allowInterrupt

Determines whether the user can interrupt the start message while it is playing.

Type: Boolean

Required: No

#### delayInSeconds

The delay between when the Lambda fulfillment function starts running and the start message is played. If the Lambda function returns before the delay is over, the start message isn't played.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 900.

Required: Yes

#### messageGroups

One to 5 message groups that contain start messages. Amazon Lex chooses one of the messages to play to the user.

Type: Array of [MessageGroup \(p. 833\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FulfillmentUpdateResponseSpecification

Service: Amazon Lex Model Building V2

Provides settings for a message that is sent periodically to the user while a fulfillment Lambda function is running.

### Contents

#### allowInterrupt

Determines whether the user can interrupt an update message while it is playing.

Type: Boolean

Required: No

#### frequencyInSeconds

The frequency that a message is sent to the user. When the period ends, Amazon Lex chooses a message from the message groups and plays it to the user. If the fulfillment Lambda returns before the first period ends, an update message is not played to the user.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 900.

Required: Yes

#### messageGroups

One to 5 message groups that contain update messages. Amazon Lex chooses one of the messages to play to the user.

Type: Array of [MessageGroup \(p. 833\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FulfillmentUpdatesSpecification

Service: Amazon Lex Model Building V2

Provides information for updating the user on the progress of fulfilling an intent.

### Contents

#### active

Determines whether fulfillment updates are sent to the user. When this field is true, updates are sent.

If the active field is set to true, the startResponse, updateResponse, and timeoutInSeconds fields are required.

Type: Boolean

Required: Yes

#### startResponse

Provides configuration information for the message sent to users when the fulfillment Lambda functions starts running.

Type: [FulfillmentStartResponseSpecification \(p. 805\)](#) object

Required: No

#### timeoutInSeconds

The length of time that the fulfillment Lambda function should run before it times out.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 900.

Required: No

#### updateResponse

Provides configuration information for messages sent periodically to the user while the fulfillment Lambda function is running.

Type: [FulfillmentUpdateResponseSpecification \(p. 806\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## GrammarSlotTypeSetting

Service: Amazon Lex Model Building V2

Settings required for a slot type based on a grammar that you provide.

### Contents

#### source

The source of the grammar used to create the slot type.

Type: [GrammarSlotTypeSource \(p. 809\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## GrammarSlotTypeSource

Service: Amazon Lex Model Building V2

Describes the Amazon S3 bucket name and location for the grammar that is the source for the slot type.

### Contents

#### kmsKeyArn

The Amazon KMS key required to decrypt the contents of the grammar, if any.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: ^arn:[\w-]+:kms:[\w-]+:[\d]{12}:(?:key\/[\w-]+\|alias\/[a-zA-Z0-9:\v\_-]{1,256})\$

Required: No

#### s3BucketName

The name of the S3 bucket that contains the grammar source.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 63.

Pattern: ^[a-zA-Z][\.\-\a-zA-Z]{1,61}[a-zA-Z]\$

Required: Yes

#### s3ObjectKey

The path to the grammar in the S3 bucket.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: [\.\-\!\\*\\_\'\(\)\a-zA-Z0-9][\.\-\!\\*\\_\'\(\)\a-zA-Z0-9]\*\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ImageResponseCard

Service: Amazon Lex Model Building V2

A card that is shown to the user by a messaging platform. You define the contents of the card, the card is displayed by the platform.

When you use a response card, the response from the user is constrained to the text associated with a button on the card.

### Contents

#### buttons

A list of buttons that should be displayed on the response card. The arrangement of the buttons is determined by the platform that displays the button.

Type: Array of [Button \(p. 774\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

Required: No

#### imageUrl

The URL of an image to display on the response card. The image URL must be publicly available so that the platform displaying the response card has access to the image.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 250.

Required: No

#### subtitle

The subtitle to display on the response card. The format of the subtitle is determined by the platform displaying the response card.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 250.

Required: No

#### title

The title to display on the response card. The format of the title is determined by the platform displaying the response card.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 250.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ImportFilter

Service: Amazon Lex Model Building V2

Filters the response from the [ListImports](#) operation.

### Contents

#### name

The name of the field to use for filtering.

Type: String

Valid Values: ImportResourceType

Required: Yes

#### operator

The operator to use for the filter. Specify EQ when the `ListImports` operation should return only resource types that equal the specified value. Specify CO when the `ListImports` operation should return resource types that contain the specified value.

Type: String

Valid Values: CO | EQ

Required: Yes

#### values

The values to use to filter the response. The values must be Bot, BotLocale, or CustomVocabulary.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[0-9a-zA-Z\_()\\s-]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ImportResourceSpecification

Service: Amazon Lex Model Building V2

Provides information about the bot or bot locale that you want to import. You can specify the `botImportSpecification` or the `botLocaleImportSpecification`, but not both.

### Contents

#### **botImportSpecification**

Parameters for importing a bot.

Type: [BotImportSpecification \(p. 750\)](#) object

Required: No

#### **botLocaleImportSpecification**

Parameters for importing a bot locale.

Type: [BotLocaleImportSpecification \(p. 755\)](#) object

Required: No

#### **customVocabularyImportSpecification**

Provides the parameters required for importing a custom vocabulary.

Type: [CustomVocabularyImportSpecification \(p. 785\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ImportSortBy

Service: Amazon Lex Model Building V2

Provides information for sorting a list of imports.

### Contents

#### attribute

The export field to use for sorting.

Type: String

Valid Values: LastUpdatedDateTime

Required: Yes

#### order

The order to sort the list.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ImportSummary

Service: Amazon Lex Model Building V2

Provides summary information about an import in an import list.

### Contents

#### **creationDateTime**

The date and time that the import was created.

Type: Timestamp

Required: No

#### **importedResourceId**

The unique identifier that Amazon Lex assigned to the imported resource.

Type: String

Length Constraints: Minimum length of 5. Maximum length of 10.

Pattern: ^([0-9a-zA-Z])+\$

Required: No

#### **importedResourceName**

The name that you gave the imported resource.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### **importedResourceType**

The type of resource that was imported.

Type: String

Valid Values: Bot | BotLocale | CustomVocabulary

Required: No

#### **importId**

The unique identifier that Amazon Lex assigned to the import.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **importStatus**

The status of the resource. When the status is Completed the resource is ready to build.

Type: String

Valid Values: InProgress | Completed | Failed | Deleting

Required: No

**lastUpdatedDateTime**

The date and time that the import was last updated.

Type: Timestamp

Required: No

**mergeStrategy**

The strategy used to merge existing bot or bot locale definitions with the imported definition.

Type: String

Valid Values: Overwrite | FailOnConflict | Append

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## InitialResponseSetting

Service: Amazon Lex Model Building V2

Configuration setting for a response sent to the user before Amazon Lex starts eliciting slots.

### Contents

#### codeHook

Settings that specify the dialog code hook that is called by Amazon Lex at a step of the conversation.

Type: [DialogCodeHookInvocationSetting \(p. 791\)](#) object

Required: No

#### conditional

Provides a list of conditional branches. Branches are evaluated in the order that they are entered in the list. The first branch with a condition that evaluates to true is executed. The last branch in the list is the default branch. The default branch should not have any condition expression. The default branch is executed if no other branch has a matching condition.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### initialResponse

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### nextStep

The next step in the conversation.

Type: [DialogState \(p. 793\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## InputContext

Service: Amazon Lex Model Building V2

The name of a context that must be active for an intent to be selected by Amazon Lex.

### Contents

#### **name**

The name of the context.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentClosingSetting

Service: Amazon Lex Model Building V2

Provides a statement the Amazon Lex conveys to the user when the intent is successfully fulfilled.

### Contents

#### active

Specifies whether an intent's closing response is used. When this field is false, the closing response isn't sent to the user. If the active field isn't specified, the default is true.

Type: Boolean

Required: No

#### closingResponse

The response that Amazon Lex sends to the user when the intent is complete.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### conditional

A list of conditional branches associated with the intent's closing response. These branches are executed when the nextStep attribute is set to EvaluateConditional.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### nextStep

Specifies the next step that the bot executes after playing the intent's closing response.

Type: [DialogState \(p. 793\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentConfirmationSetting

Service: Amazon Lex Model Building V2

Provides a prompt for making sure that the user is ready for the intent to be fulfilled.

### Contents

#### active

Specifies whether the intent's confirmation is sent to the user. When this field is false, confirmation and declination responses aren't sent. If the active field isn't specified, the default is true.

Type: Boolean

Required: No

#### codeHook

The DialogCodeHookInvocationSetting object associated with intent's confirmation step. The dialog code hook is triggered based on these invocation settings when the confirmation next step or declination next step or failure next step is InvokeDialogCodeHook.

Type: [DialogCodeHookInvocationSetting \(p. 791\)](#) object

Required: No

#### confirmationConditional

A list of conditional branches to evaluate after the intent is closed.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### confirmationNextStep

Specifies the next step that the bot executes when the customer confirms the intent.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### confirmationResponse

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### declinationConditional

A list of conditional branches to evaluate after the intent is declined.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### declinationNextStep

Specifies the next step that the bot executes when the customer declines the intent.

Type: [DialogState \(p. 793\)](#) object

Required: No

### **declinationResponse**

When the user answers "no" to the question defined in `promptSpecification`, Amazon Lex responds with this response to acknowledge that the intent was canceled.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

### **elicitationCodeHook**

The `DialogCodeHookInvocationSetting` used when the code hook is invoked during confirmation prompt retries.

Type: [ElicitationCodeHookInvocationSetting \(p. 795\)](#) object

Required: No

### **failureConditional**

Provides a list of conditional branches. Branches are evaluated in the order that they are entered in the list. The first branch with a condition that evaluates to true is executed. The last branch in the list is the default branch. The default branch should not have any condition expression. The default branch is executed if no other branch has a matching condition.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

### **failureNextStep**

The next step to take in the conversation if the confirmation step fails.

Type: [DialogState \(p. 793\)](#) object

Required: No

### **failureResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

### **promptSpecification**

Prompts the user to confirm the intent. This question should have a yes or no answer.

Amazon Lex uses this prompt to ensure that the user acknowledges that the intent is ready for fulfillment. For example, with the `OrderPizza` intent, you might want to confirm that the order is correct before placing it. For other intents, such as intents that simply respond to user questions, you might not need to ask the user for confirmation before providing the information.

Type: [PromptSpecification \(p. 846\)](#) object

Required: Yes

## **See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentFilter

Service: Amazon Lex Model Building V2

Filters the response from the `ListIntents` operation.

### Contents

#### **name**

The name of the field to use for the filter.

Type: String

Valid Values: IntentName

Required: Yes

#### **operator**

The operator to use for the filter. Specify EQ when the `ListIntents` operation should return only aliases that equal the specified value. Specify CO when the `ListIntents` operation should return aliases that contain the specified value.

Type: String

Valid Values: CO | EQ

Required: Yes

#### **values**

The value to use for the filter.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[0-9a-zA-Z\_()\\s-]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentOverride

Service: Amazon Lex Model Building V2

Override settings to configure the intent state.

### Contents

#### **name**

The name of the intent. Only required when you're switching intents.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### **slots**

A map of all of the slot value overrides for the intent. The name of the slot maps to the value of the slot. Slots that are not included in the map aren't overridden.,

Type: String to [SlotValueOverride \(p. 875\)](#) object map

Key Length Constraints: Minimum length of 1. Maximum length of 100.

Key Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of intents.

### Contents

#### attribute

The attribute to use to sort the list of intents.

Type: String

Valid Values: IntentName | LastUpdatedDateTime

Required: Yes

#### order

The order to sort the list. You can choose ascending or descending.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentStatistics

Service: Amazon Lex Model Building V2

The object that contains the statistical summary of recommended intents associated with the bot recommendation.

### Contents

#### discoveredIntentCount

The number of recommended intents associated with the bot recommendation.

Type: Integer

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentSummary

Service: Amazon Lex Model Building V2

Summary information about an intent returned by the `ListIntents` operation.

### Contents

#### **description**

The description of the intent.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **inputContexts**

The input contexts that must be active for this intent to be considered for recognition.

Type: Array of [InputContext \(p. 818\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

Required: No

#### **intentId**

The unique identifier assigned to the intent. Use this ID to get detailed information about the intent with the `DescribeIntent` operation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **intentName**

The name of the intent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### **lastUpdatedDateTime**

The timestamp of the date and time that the intent was last updated.

Type: Timestamp

Required: No

#### **outputContexts**

The output contexts that are activated when this intent is fulfilled.

Type: Array of [OutputContext \(p. 837\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

**parentIntentSignature**

If this intent is derived from a built-in intent, the name of the parent intent.

Type: String

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## KendraConfiguration

Service: Amazon Lex Model Building V2

Provides configuration information for the AMAZON.KendraSearchIntent intent. When you use this intent, Amazon Lex searches the specified Amazon Kendra index and returns documents from the index that match the user's utterance.

### Contents

#### kendraIndex

The Amazon Resource Name (ARN) of the Amazon Kendra index that you want the AMAZON.KendraSearchIntent intent to search. The index must be in the same account and Region as the Amazon Lex bot.

Type: String

Length Constraints: Minimum length of 32. Maximum length of 2048.

Pattern: ^arn:aws:kendra:[a-z]+-[a-z]+-[0-9]:[0-9]{12}:index\/[a-zA-Z0-9\_-]\*\$

Required: Yes

#### queryFilterString

A query filter that Amazon Lex sends to Amazon Kendra to filter the response from a query. The filter is in the format defined by Amazon Kendra. For more information, see [Filtering queries](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5000.

Required: No

#### queryFilterStringEnabled

Determines whether the AMAZON.KendraSearchIntent intent uses a custom query string to query the Amazon Kendra index.

Type: Boolean

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## LambdaCodeHook

Service: Amazon Lex Model Building V2

Specifies a Lambda function that verifies requests to a bot or fulfills the user's request to a bot.

### Contents

#### codeHookInterfaceVersion

The version of the request-response that you want Amazon Lex to use to invoke your Lambda function.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5.

Required: Yes

#### lambdaARN

The Amazon Resource Name (ARN) of the Lambda function.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:aws:lambda:[a-z]+-[a-z]+-[0-9]:[0-9]{12}:function:[a-zA-Z0-9-_]+(/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})?(:[a-zA-Z0-9-_]+)?`

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## LexTranscriptFilter

Service: Amazon Lex Model Building V2

The object that contains transcript filter details that are associated with a bot recommendation.

### Contents

#### **dateRangeFilter**

The object that contains a date range filter that will be applied to the transcript. Specify this object if you want Amazon Lex to only read the files that are within the date range.

Type: [DateRangeFilter \(p. 788\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Message

Service: Amazon Lex Model Building V2

The object that provides message text and its type.

### Contents

#### **customPayload**

A message in a custom format defined by the client application.

Type: [CustomPayload \(p. 782\)](#) object

Required: No

#### **imageResponseCard**

A message that defines a response card that the client application can show to the user.

Type: [ImageResponseCard \(p. 810\)](#) object

Required: No

#### **plainTextMessage**

A message in plain text format.

Type: [PlainTextMessage \(p. 839\)](#) object

Required: No

#### **ssmlMessage**

A message in Speech Synthesis Markup Language (SSML).

Type: [SSMLMessage \(p. 879\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## MessageGroup

Service: Amazon Lex Model Building V2

Provides one or more messages that Amazon Lex should send to the user.

### Contents

#### message

The primary message that Amazon Lex should send to the user.

Type: [Message \(p. 832\)](#) object

Required: Yes

#### variations

Message variations to send to the user. When variations are defined, Amazon Lex chooses the primary message or one of the variations to send to the user.

Type: Array of [Message \(p. 832\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 2 items.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## MultipleValuesSetting

Service: Amazon Lex Model Building V2

Indicates whether a slot can return multiple values.

### Contents

#### allowMultipleValues

Indicates whether a slot can return multiple values. When `true`, the slot may return more than one value in a response. When `false`, the slot returns only a single value.

Multi-value slots are only available in the en-US locale. If you set this value to `true` in any other locale, Amazon Lex throws a `ValidationException`.

If the `allowMutlipleValues` is not set, the default value is `false`.

Type: Boolean

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## NewCustomVocabularyItem

Service: Amazon Lex Model Building V2

The new custom vocabulary item from the custom vocabulary list.

### Contents

#### displayAs

The display as value assigned to the new custom vocabulary item from the custom vocabulary list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: No

#### phrase

The unique phrase for the new custom vocabulary item from the custom vocabulary list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: Yes

#### weight

The weight assigned to the new custom vocabulary item from the custom vocabulary list.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 3.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ObfuscationSetting

Service: Amazon Lex Model Building V2

Determines whether Amazon Lex obscures slot values in conversation logs.

### Contents

#### obfuscationSettingType

Value that determines whether Amazon Lex obscures slot values in conversation logs. The default is to obscure the values.

Type: String

Valid Values: None | DefaultObfuscation

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## OutputContext

Service: Amazon Lex Model Building V2

Describes a session context that is activated when an intent is fulfilled.

### Contents

#### **name**

The name of the output context.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### **timeToLiveInSeconds**

The amount of time, in seconds, that the output context should remain active. The time is figured from the first time the context is sent to the user.

Type: Integer

Valid Range: Minimum value of 5. Maximum value of 86400.

Required: Yes

#### **turnsToLive**

The number of conversation turns that the output context should remain active. The number of turns is counted from the first time that the context is sent to the user.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 20.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PathFormat

Service: Amazon Lex Model Building V2

The object that contains a path format that will be applied when Amazon Lex reads the transcript file in the bucket you provide. Specify this object if you only want Lex to read a subset of files in your Amazon S3 bucket.

### Contents

#### objectPrefixes

A list of Amazon S3 prefixes that points to sub-folders in the Amazon S3 bucket. Specify this list if you only want Lex to read the files under this set of sub-folders.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Length Constraints: Minimum length of 1.

Pattern: `^[\/\]?+[a-zA-Z0-9!_.*'()]-]+(\/[a-zA-Z0-9!_.*'()]-)+*$`

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PlainTextMessage

Service: Amazon Lex Model Building V2

Defines an ASCII text message to send to the user.

### Contents

#### **value**

The message to send to the user.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PostDialogCodeHookInvocationSpecification

Service: Amazon Lex Model Building V2

Specifies next steps to run after the dialog code hook finishes.

### Contents

#### **failureConditional**

A list of conditional branches to evaluate after the dialog code hook throws an exception or returns with the State field of the Intent object set to Failed.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### **failureNextStep**

Specifies the next step the bot runs after the dialog code hook throws an exception or returns with the State field of the Intent object set to Failed.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### **failureResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### **successConditional**

A list of conditional branches to evaluate after the dialog code hook finishes successfully.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### **successNextStep**

Specifies the next step the bot runs after the dialog code hook finishes successfully.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### **successResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### **timeoutConditional**

A list of conditional branches to evaluate if the code hook times out.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

**timeoutNextStep**

Specifies the next step that the bot runs when the code hook times out.

Type: [DialogState \(p. 793\)](#) object

Required: No

**timeoutResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PostFulfillmentStatusSpecification

Service: Amazon Lex Model Building V2

Provides a setting that determines whether the post-fulfillment response is sent to the user. For more information, see <https://docs.aws.amazon.com/lexv2/latest/dg/streaming-progress.html#progress-complete>

### Contents

#### **failureConditional**

A list of conditional branches to evaluate after the fulfillment code hook throws an exception or returns with the State field of the Intent object set to Failed.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### **failureNextStep**

Specifies the next step the bot runs after the fulfillment code hook throws an exception or returns with the State field of the Intent object set to Failed.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### **failureResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### **successConditional**

A list of conditional branches to evaluate after the fulfillment code hook finishes successfully.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### **successNextStep**

Specifies the next step in the conversation that Amazon Lex invokes when the fulfillment code hook completes successfully.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### **successResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### **timeoutConditional**

A list of conditional branches to evaluate if the fulfillment code hook times out.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

**timeoutNextStep**

Specifies the next step that the bot runs when the fulfillment code hook times out.

Type: [DialogState \(p. 793\)](#) object

Required: No

**timeoutResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Principal

Service: Amazon Lex Model Building V2

The IAM principal that you allowing or denying access to an Amazon Lex action. You must provide a service or an arn, but not both in the same statement. For more information, see [AWS JSON policy elements: Principal](#).

### Contents

#### arn

The Amazon Resource Name (ARN) of the principal.

Type: String

Length Constraints: Minimum length of 30. Maximum length of 1024.

Pattern: ^arn:aws:iam::[0-9]{12}:(root|(user|role)/.\*)\$

Required: No

#### service

The name of the AWS service that should allowed or denied access to an Amazon Lex action.

Type: String

Length Constraints: Minimum length of 15. Maximum length of 1024.

Pattern: ^[0-9a-zA-Z\_.]+\$

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PromptAttemptSpecification

Service: Amazon Lex Model Building V2

Specifies the settings on a prompt attempt.

### Contents

#### **allowedInputTypes**

Indicates the allowed input types of the prompt attempt.

Type: [AllowedInputTypes \(p. 737\)](#) object

Required: Yes

#### **allowInterrupt**

Indicates whether the user can interrupt a speech prompt attempt from the bot.

Type: Boolean

Required: No

#### **audioAndDTMFInputSpecification**

Specifies the settings on audio and DTMF input.

Type: [AudioAndDTMFInputSpecification \(p. 740\)](#) object

Required: No

#### **textInputSpecification**

Specifies the settings on text input.

Type: [TextInputSpecification \(p. 884\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PromptSpecification

Service: Amazon Lex Model Building V2

Specifies a list of message groups that Amazon Lex sends to a user to elicit a response.

### Contents

#### allowInterrupt

Indicates whether the user can interrupt a speech prompt from the bot.

Type: Boolean

Required: No

#### maxRetries

The maximum number of times the bot tries to elicit a response from the user using this prompt.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 5.

Required: Yes

#### messageGroups

A collection of messages that Amazon Lex can send to the user. Amazon Lex chooses the actual message to send at runtime.

Type: Array of [MessageGroup \(p. 833\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Required: Yes

#### messageSelectionStrategy

Indicates how a message is selected from a message group among retries.

Type: String

Valid Values: Random | Ordered

Required: No

#### promptAttemptsSpecification

Specifies the advanced settings on each attempt of the prompt.

Type: String to [PromptAttemptSpecification \(p. 845\)](#) object map

Valid Keys: Initial | Retry1 | Retry2 | Retry3 | Retry4 | Retry5

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## RecommendedIntentSummary

Service: Amazon Lex Model Building V2

An object that contains a summary of a recommended intent.

### Contents

#### **intentId**

The unique identifier of a recommended intent associated with the bot recommendation.

Type: String

Length Constraints: Fixed length of 10.

Pattern: `^[0-9a-zA-Z]+$`

Required: No

#### **intentName**

The name of a recommended intent associated with the bot recommendation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^([0-9a-zA-Z][_-]?)+$`

Required: No

#### **sampleUtterancesCount**

The count of sample utterances of a recommended intent that is associated with a bot recommendation.

Type: Integer

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## RelativeAggregationDuration

Service: Amazon Lex Model Building V2

Specifies the time window that utterance statistics are returned for. The time window is always relative to the last time that the utterances were aggregated. For example, if the `ListAggregatedUtterances` operation is called at 1600, the time window is set to 1 hour, and the last refresh time was 1530, only utterances made between 1430 and 1530 are returned.

You can choose the time window that statistics should be returned for.

- **Hours** - You can request utterance statistics for 1, 3, 6, 12, or 24 hour time windows. Statistics are refreshed every half hour for 1 hour time windows, and hourly for the other time windows.
- **Days** - You can request utterance statistics for 3 days. Statistics are refreshed every 6 hours.
- **Weeks** - You can see statistics for one or two weeks. Statistics are refreshed every 12 hours for one week time windows, and once per day for two week time windows.

## Contents

### timeDimension

The type of time period that the `timeValue` field represents.

Type: String

Valid Values: Hours | Days | Weeks

Required: Yes

### timeValue

The period of the time window to gather statistics for. The valid value depends on the setting of the `timeDimension` field.

- Hours - 1/3/6/12/24
- Days - 3
- Weeks - 1/2

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 24.

Required: Yes

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ResponseSpecification

Service: Amazon Lex Model Building V2

Specifies a list of message groups that Amazon Lex uses to respond the user input.

### Contents

#### allowInterrupt

Indicates whether the user can interrupt a speech response from Amazon Lex.

Type: Boolean

Required: No

#### messageGroups

A collection of responses that Amazon Lex can send to the user. Amazon Lex chooses the actual response to send at runtime.

Type: Array of [MessageGroup \(p. 833\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## S3BucketLogDestination

Service: Amazon Lex Model Building V2

Specifies an Amazon S3 bucket for logging audio conversations

### Contents

#### kmsKeyArn

The Amazon Resource Name (ARN) of an AWS Key Management Service (KMS) key for encrypting audio log files stored in an S3 bucket.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: ^arn:[\w\ -]+:kms:[\w\ -]+:[\d]{12}:(?:key\/[\w\ -]+\|alias\/[a-zA-Z0-9:\ \\_\ -]\{1,256\})\$

Required: No

#### logPrefix

The S3 prefix to assign to audio log files.

Type: String

Length Constraints: Maximum length of 1024.

Required: Yes

#### s3BucketArn

The Amazon Resource Name (ARN) of an Amazon S3 bucket where audio log files are stored.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^arn:[\w\ -]+:s3:::[a-zA-Z0-9][\.\-\w\ -]{1,61}[a-zA-Z0-9]\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## S3BucketTranscriptSource

Service: Amazon Lex Model Building V2

The object representing the Amazon S3 bucket containing the transcript, as well as the associated metadata.

### Contents

#### kmsKeyArn

The ARN of the KMS key that customer use to encrypt their Amazon S3 bucket. Only use this field if your bucket is encrypted using a customer managed KMS key.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: ^arn:[\w-]+:kms:[\w-]+:[\d]{12}:(?:key\/[\w-]+|alias\/[a-zA-Z0-9:\v\_-]{1,256})\$

Required: No

#### pathFormat

The object that contains a path format that will be applied when Amazon Lex reads the transcript file in the bucket you provide. Specify this object if you only want Lex to read a subset of files in your Amazon S3 bucket.

Type: [PathFormat \(p. 838\)](#) object

Required: No

#### s3BucketName

The name of the bucket containing the transcript and the associated metadata.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 63.

Pattern: ^[a-zA-Z0-9][\.-a-zA-Z0-9]{1,61}[a-zA-Z0-9]\$

Required: Yes

#### transcriptFilter

The object that contains the filter which will be applied when Amazon Lex reads through the Amazon S3 bucket. Specify this object if you want Amazon Lex to read only a subset of the Amazon S3 bucket based on the filter you provide.

Type: [TranscriptFilter \(p. 887\)](#) object

Required: No

#### transcriptFormat

The format of the transcript content. Currently, Genie only supports the Amazon Lex transcript format.

Type: String

Valid Values: Lex

Required: Yes

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SampleUtterance

Service: Amazon Lex Model Building V2

A sample utterance that invokes an intent or respond to a slot elicitation prompt.

### Contents

#### **utterance**

The sample utterance that Amazon Lex uses to build its machine-learning model to recognize intents.

Type: String

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SampleValue

Service: Amazon Lex Model Building V2

Defines one of the values for a slot type.

### Contents

#### **value**

The value that can be used for a slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 140.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SentimentAnalysisSettings

Service: Amazon Lex Model Building V2

Determines whether Amazon Lex will use Amazon Comprehend to detect the sentiment of user utterances.

### Contents

#### **detectSentiment**

Sets whether Amazon Lex uses Amazon Comprehend to detect the sentiment of user utterances.

Type: Boolean

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotCaptureSetting

Service: Amazon Lex Model Building V2

Settings used when Amazon Lex successfully captures a slot value from a user.

### Contents

#### **captureConditional**

A list of conditional branches to evaluate after the slot value is captured.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### **captureNextStep**

Specifies the next step that the bot runs when the slot value is captured before the code hook times out.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### **captureResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

#### **codeHook**

Code hook called after Amazon Lex successfully captures a slot value.

Type: [DialogCodeHookInvocationSetting \(p. 791\)](#) object

Required: No

#### **elicitationCodeHook**

Code hook called when Amazon Lex doesn't capture a slot value.

Type: [ElicitationCodeHookInvocationSetting \(p. 795\)](#) object

Required: No

#### **failureConditional**

A list of conditional branches to evaluate when the slot value isn't captured.

Type: [ConditionalSpecification \(p. 780\)](#) object

Required: No

#### **failureNextStep**

Specifies the next step that the bot runs when the slot value code is not recognized.

Type: [DialogState \(p. 793\)](#) object

Required: No

#### **failureResponse**

Specifies a list of message groups that Amazon Lex uses to respond the user input.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotDefaultValue

Service: Amazon Lex Model Building V2

Specifies the default value to use when a user doesn't provide a value for a slot.

### Contents

#### defaultValue

The default value to use when a user doesn't provide a value for a slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 202.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotDefaultValueSpecification

Service: Amazon Lex Model Building V2

Defines a list of values that Amazon Lex should use as the default value for a slot.

### Contents

#### defaultValueList

A list of default values. Amazon Lex chooses the default value to use in the order that they are presented in the list.

Type: Array of [SlotDefaultValue \(p. 859\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotFilter

Service: Amazon Lex Model Building V2

Filters the response from the `ListSlots` operation.

### Contents

#### **name**

The name of the field to use for filtering.

Type: String

Valid Values: SlotName

Required: Yes

#### **operator**

The operator to use for the filter. Specify EQ when the `ListSlots` operation should return only aliases that equal the specified value. Specify CO when the `ListSlots` operation should return aliases that contain the specified value.

Type: String

Valid Values: CO | EQ

Required: Yes

#### **values**

The value to use to filter the response.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[0-9a-zA-Z\_()\\s-]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotPriority

Service: Amazon Lex Model Building V2

Sets the priority that Amazon Lex should use when eliciting slot values from a user.

### Contents

#### priority

The priority that a slot should be elicited.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100.

Required: Yes

#### slotId

The unique identifier of the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of bots.

### Contents

#### attribute

The attribute to use to sort the list.

Type: String

Valid Values: SlotName | LastUpdatedDateTime

Required: Yes

#### order

The order to sort the list. You can choose ascending or descending.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotSummary

Service: Amazon Lex Model Building V2

Summary information about a slot, a value that the bot elicits from the user.

### Contents

#### **description**

The description of the slot.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **lastUpdatedDateTime**

The timestamp of the last date and time that the slot was updated.

Type: Timestamp

Required: No

#### **slotConstraint**

Whether the slot is required or optional. An intent is complete when all required slots are filled.

Type: String

Valid Values: Required | Optional

Required: No

#### **slotId**

The unique identifier of the slot.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **slotName**

The name given to the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

#### **slotTypeId**

The unique identifier for the slot type that defines the values for the slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern: ^((AMAZON\\.)[a-zA-Z\_]+?)|[0-9a-zA-Z]+)\$

Required: No

#### **valueElicitationPromptSpecification**

Prompts that are sent to the user to elicit a value for the slot.

Type: [PromptSpecification \(p. 846\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotTypeFilter

Service: Amazon Lex Model Building V2

Filters the response from the `ListSlotTypes` operation.

### Contents

#### **name**

The name of the field to use for filtering.

Type: String

Valid Values: `SlotTypeName` | `ExternalSourceType`

Required: Yes

#### **operator**

The operator to use for the filter. Specify `EQ` when the `ListSlotTypes` operation should return only aliases that equal the specified value. Specify `CO` when the `ListSlotTypes` operation should return aliases that contain the specified value.

Type: String

Valid Values: `CO` | `EQ`

Required: Yes

#### **values**

The value to use to filter the response.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\0-9a-zA-Z_()\\s-]+$`

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotTypeSortBy

Service: Amazon Lex Model Building V2

Specifies attributes for sorting a list of slot types.

### Contents

#### attribute

The attribute to use to sort the list of slot types.

Type: String

Valid Values: SlotTypeName | LastUpdatedDateTime

Required: Yes

#### order

The order to sort the list. You can say ascending or descending.

Type: String

Valid Values: Ascending | Descending

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotTypeStatistics

Service: Amazon Lex Model Building V2

The object that contains the statistical summary of the recommended slot type associated with the bot recommendation.

### Contents

#### discoveredSlotTypeCount

The number of recommended slot types associated with the bot recommendation.

Type: Integer

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotTypeSummary

Service: Amazon Lex Model Building V2

Provides summary information about a slot type.

### Contents

#### **description**

The description of the slot type.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 200.

Required: No

#### **lastUpdatedDateTime**

A timestamp of the date and time that the slot type was last updated.

Type: Timestamp

Required: No

#### **parentSlotTypeSignature**

If the slot type is derived from a built-on slot type, the name of the parent slot type.

Type: String

Required: No

#### **slotTypeCategory**

Indicates the type of the slot type.

- Custom - A slot type that you created using custom values. For more information, see [Creating custom slot types](#).
- Extended - A slot type created by extending the AMAZON.AlphaNumeric built-in slot type. For more information, see [AMAZON.AlphaNumeric](#).
- ExternalGrammar - A slot type using a custom GRXML grammar to define values. For more information, see [Using a custom grammar slot type](#).

Type: String

Valid Values: Custom | Extended | ExternalGrammar | Composite

Required: No

#### **slotTypeId**

The unique identifier assigned to the slot type.

Type: String

Length Constraints: Fixed length of 10.

Pattern: ^[0-9a-zA-Z]+\$

Required: No

#### **slotTypeName**

The name of the slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotTypeValue

Service: Amazon Lex Model Building V2

Each slot type can have a set of values. Each SlotTypeValue represents a value that the slot type can take.

### Contents

#### **sampleValue**

The value of the slot type entry.

Type: [SampleValue \(p. 855\)](#) object

Required: No

#### **synonyms**

Additional values related to the slot type entry.

Type: Array of [SampleValue \(p. 855\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10000 items.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotValue

Service: Amazon Lex Model Building V2

The value to set in a slot.

### Contents

#### interpretedValue

The value that Amazon Lex determines for the slot. The actual value depends on the setting of the value selection strategy for the bot. You can choose to use the value entered by the user, or you can have Amazon Lex choose the first value in the `resolvedValues` list.

Type: String

Length Constraints: Minimum length of 1.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotValueElicitationSetting

Service: Amazon Lex Model Building V2

Specifies the elicitation setting details for constituent sub slots of a composite slot.

### Contents

#### defaultValueSpecification

A list of default values for a slot. Default values are used when Amazon Lex hasn't determined a value for a slot. You can specify default values from context variables, session attributes, and defined values.

Type: [SlotDefaultValueSpecification \(p. 860\)](#) object

Required: No

#### promptSpecification

The prompt that Amazon Lex uses to elicit the slot value from the user.

Type: [PromptSpecification \(p. 846\)](#) object

Required: No

#### sampleUtterances

If you know a specific pattern that users might respond to an Amazon Lex request for a slot value, you can provide those utterances to improve accuracy. This is optional. In most cases, Amazon Lex is capable of understanding user utterances.

Type: Array of [SampleUtterance \(p. 854\)](#) objects

Required: No

#### slotCaptureSetting

Specifies the settings that Amazon Lex uses when a slot value is successfully entered by a user.

Type: [SlotCaptureSetting \(p. 857\)](#) object

Required: No

#### slotConstraint

Specifies whether the slot is required or optional.

Type: String

Valid Values: Required | Optional

Required: Yes

#### waitAndContinueSpecification

Specifies the prompts that Amazon Lex uses while a bot is waiting for customer input.

Type: [WaitAndContinueSpecification \(p. 891\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotValueOverride

Service: Amazon Lex Model Building V2

The slot values that Amazon Lex uses when it sets slot values in a dialog step.

### Contents

#### shape

When the shape value is `List`, it indicates that the `values` field contains a list of slot values. When the value is `Scalar`, it indicates that the `value` field contains a single value.

Type: String

Valid Values: `Scalar` | `List`

Required: No

#### value

The current value of the slot.

Type: [SlotValue \(p. 872\)](#) object

Required: No

#### values

A list of one or more values that the user provided for the slot. For example, for a slot that elicits pizza toppings, the values might be "pepperoni" and "pineapple."

Type: Array of [SlotValueOverride \(p. 875\)](#) objects

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotValueRegexFilter

Service: Amazon Lex Model Building V2

Provides a regular expression used to validate the value of a slot.

### Contents

#### pattern

A regular expression used to validate the value of a slot.

Use a standard regular expression. Amazon Lex supports the following characters in the regular expression:

- A-Z, a-z
- 0-9
- Unicode characters ("\\ u<Unicode>")

Represent Unicode characters with four digits, for example "\u0041" or "\u005A".

The following regular expression operators are not supported:

- Infinite repeaters: \*, +, or {x,} with no upper bound.
- Wild card (.)

Type: String

Length Constraints: Minimum length of 1. Maximum length of 300.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SlotValueSelectionSetting

Service: Amazon Lex Model Building V2

Contains settings used by Amazon Lex to select a slot value.

### Contents

#### advancedRecognitionSetting

Provides settings that enable advanced recognition settings for slot values.

Type: [AdvancedRecognitionSetting \(p. 732\)](#) object

Required: No

#### regexFilter

A regular expression used to validate the value of a slot.

Type: [SlotValueRegexFilter \(p. 876\)](#) object

Required: No

#### resolutionStrategy

Determines the slot resolution strategy that Amazon Lex uses to return slot type values. The field can be set to one of the following values:

- OriginalValue - Returns the value entered by the user, if the user value is similar to the slot value.
- TopResolution - If there is a resolution list for the slot, return the first value in the resolution list as the slot type value. If there is no resolution list, null is returned.

If you don't specify the valueSelectionStrategy, the default is OriginalValue.

Type: String

Valid Values: OriginalValue | TopResolution | Concatenation

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Specifications

Service: Amazon Lex Model Building V2

Subslot specifications.

### Contents

#### **slotTypeId**

The unique identifier assigned to the slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern: ^((AMAZON\\.)[a-zA-Z\_]+?)|[0-9a-zA-Z]+)\$

Required: Yes

#### **valueElicitationSetting**

Specifies the elicitation setting details for constituent sub slots of a composite slot.

Type: [SubSlotValueElicitationSetting \(p. 883\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SSMLMessage

Service: Amazon Lex Model Building V2

Defines a Speech Synthesis Markup Language (SSML) prompt.

### Contents

#### **value**

The SSML text that defines the prompt.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## StillWaitingResponseSpecification

Service: Amazon Lex Model Building V2

Defines the messages that Amazon Lex sends to a user to remind them that the bot is waiting for a response.

### Contents

#### allowInterrupt

Indicates that the user can interrupt the response by speaking while the message is being played.

Type: Boolean

Required: No

#### frequencyInSeconds

How often a message should be sent to the user. Minimum of 1 second, maximum of 5 minutes.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 300.

Required: Yes

#### messageGroups

One or more message groups, each containing one or more messages, that define the prompts that Amazon Lex sends to the user.

Type: Array of [MessageGroup \(p. 833\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Required: Yes

#### timeoutInSeconds

If Amazon Lex waits longer than this length of time for a response, it will stop sending messages.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 900.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SubSlotSetting

Service: Amazon Lex Model Building V2

Specifications for the constituent sub slots and the expression for the composite slot.

### Contents

#### expression

The expression text for defining the constituent sub slots in the composite slot using logical AND and OR operators.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 640.

Pattern: [0-9A-Za-z\_\-\s\(\)]<sup>+</sup>

Required: No

#### slotSpecifications

Specifications for the constituent sub slots of a composite slot.

Type: String to [Specifications \(p. 878\)](#) object map

Map Entries: Minimum number of 0 items. Maximum number of 6 items.

Key Length Constraints: Minimum length of 1. Maximum length of 100.

Key Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SubSlotTypeComposition

Service: Amazon Lex Model Building V2

Subslot type composition.

### Contents

#### **name**

Name of a constituent sub slot inside a composite slot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: Yes

#### **slotTypeId**

The unique identifier assigned to a slot type. This refers to either a built-in slot type or the unique slotTypeId of a custom slot type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 25.

Pattern: ^((AMAZON\.)[a-zA-Z\_]+?)|[0-9a-zA-Z]+)\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SubSlotValueElicitationSetting

Service: Amazon Lex Model Building V2

Subslot elicitation settings.

**DefaultValueSpecification** is a list of default values for a constituent sub slot in a composite slot. Default values are used when Amazon Lex hasn't determined a value for a slot. You can specify default values from context variables, session attributes, and defined values. This is similar to **DefaultValueSpecification** for slots.

**PromptSpecification** is the prompt that Amazon Lex uses to elicit the sub slot value from the user. This is similar to **PromptSpecification** for slots.

### Contents

#### **defaultValueSpecification**

Defines a list of values that Amazon Lex should use as the default value for a slot.

Type: [SlotDefaultValueSpecification \(p. 860\)](#) object

Required: No

#### **promptSpecification**

Specifies a list of message groups that Amazon Lex sends to a user to elicit a response.

Type: [PromptSpecification \(p. 846\)](#) object

Required: Yes

#### **sampleUtterances**

If you know a specific pattern that users might respond to an Amazon Lex request for a sub slot value, you can provide those utterances to improve accuracy. This is optional. In most cases Amazon Lex is capable of understanding user utterances. This is similar to **SampleUtterances** for slots.

Type: Array of [SampleUtterance \(p. 854\)](#) objects

Required: No

#### **waitAndContinueSpecification**

Specifies the prompts that Amazon Lex uses while a bot is waiting for customer input.

Type: [WaitAndContinueSpecification \(p. 891\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TextInputSpecification

Service: Amazon Lex Model Building V2

Specifies the text input specifications.

### Contents

#### startTimeoutMs

Time for which a bot waits before re-prompting a customer for text input.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TextLogDestination

Service: Amazon Lex Model Building V2

Defines the Amazon CloudWatch Logs destination log group for conversation text logs.

### Contents

#### cloudWatch

Defines the Amazon CloudWatch Logs log group where text and metadata logs are delivered.

Type: [CloudWatchLogGroupLogDestination \(p. 775\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TextLogSetting

Service: Amazon Lex Model Building V2

Defines settings to enable text conversation logs.

### Contents

#### destination

Defines the Amazon CloudWatch Logs destination log group for conversation text logs.

Type: [TextLogDestination \(p. 885\)](#) object

Required: Yes

#### enabled

Determines whether conversation logs should be stored for an alias.

Type: Boolean

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TranscriptFilter

Service: Amazon Lex Model Building V2

The object representing the filter that Amazon Lex will use to select the appropriate transcript.

### Contents

#### lexTranscriptFilter

The object representing the filter that Amazon Lex will use to select the appropriate transcript when the transcript format is the Amazon Lex format.

Type: [LexTranscriptFilter \(p. 831\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TranscriptSourceSetting

Service: Amazon Lex Model Building V2

Indicates the setting of the location where the transcript is stored.

### Contents

#### s3BucketTranscriptSource

Indicates the setting of the Amazon S3 bucket where the transcript is stored.

Type: [S3BucketTranscriptSource \(p. 852\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## UtteranceAggregationDuration

Service: Amazon Lex Model Building V2

Provides parameters for setting the time window and duration for aggregating utterance data.

### Contents

#### relativeAggregationDuration

The desired time window for aggregating utterances.

Type: [RelativeAggregationDuration \(p. 849\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## VoiceSettings

Service: Amazon Lex Model Building V2

Defines settings for using an Amazon Polly voice to communicate with a user.

### Contents

#### **engine**

Indicates the type of Amazon Polly voice that Amazon Lex should use for voice interaction with the user. For more information, see the [engine parameter of the SynthesizeSpeech operation](#) in the [Amazon Polly developer guide](#).

If you do not specify a value, the default is standard.

Type: String

Valid Values: standard | neural

Required: No

#### **voiceld**

The identifier of the Amazon Polly voice to use.

Type: String

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## WaitAndContinueSpecification

Service: Amazon Lex Model Building V2

Specifies the prompts that Amazon Lex uses while a bot is waiting for customer input.

### Contents

#### active

Specifies whether the bot will wait for a user to respond. When this field is false, wait and continue responses for a slot aren't used. If the active field isn't specified, the default is true.

Type: Boolean

Required: No

#### continueResponse

The response that Amazon Lex sends to indicate that the bot is ready to continue the conversation.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: Yes

#### stillWaitingResponse

A response that Amazon Lex sends periodically to the user to indicate that the bot is still waiting for input from the user.

Type: [StillWaitingResponseSpecification \(p. 880\)](#) object

Required: No

#### waitingResponse

The response that Amazon Lex sends to indicate that the bot is waiting for the conversation to continue.

Type: [ResponseSpecification \(p. 850\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Amazon Lex Runtime V2

The following data types are supported by Amazon Lex Runtime V2:

- [ActiveContext \(p. 893\)](#)
- [ActiveContextTimeToLive \(p. 895\)](#)
- [AudioInputEvent \(p. 896\)](#)

- [AudioResponseEvent \(p. 898\)](#)
- [Button \(p. 899\)](#)
- [ConfidenceScore \(p. 900\)](#)
- [ConfigurationEvent \(p. 901\)](#)
- [DialogAction \(p. 903\)](#)
- [DisconnectionEvent \(p. 905\)](#)
- [DTMFInputEvent \(p. 906\)](#)
- [ElicitSubSlot \(p. 907\)](#)
- [HeartbeatEvent \(p. 908\)](#)
- [ImageResponseCard \(p. 909\)](#)
- [Intent \(p. 911\)](#)
- [IntentResultEvent \(p. 912\)](#)
- [Interpretation \(p. 914\)](#)
- [Message \(p. 915\)](#)
- [PlaybackCompletionEvent \(p. 916\)](#)
- [PlaybackInterruptionEvent \(p. 917\)](#)
- [RuntimeHintDetails \(p. 918\)](#)
- [RuntimeHints \(p. 919\)](#)
- [RuntimeHintValue \(p. 920\)](#)
- [SentimentResponse \(p. 921\)](#)
- [SentimentScore \(p. 922\)](#)
- [SessionState \(p. 923\)](#)
- [Slot \(p. 925\)](#)
- [StartConversationRequestEventStream \(p. 926\)](#)
- [StartConversationResponseEventStream \(p. 928\)](#)
- [TextInputEvent \(p. 931\)](#)
- [TextResponseEvent \(p. 932\)](#)
- [TranscriptEvent \(p. 933\)](#)
- [Value \(p. 934\)](#)

## ActiveContext

Service: Amazon Lex Runtime V2

Contains information about the contexts that a user is using in a session. You can configure Amazon Lex V2 to set a context when an intent is fulfilled, or you can set a context using the [PutSession \(p. 698\)](#), [RecognizeText \(p. 703\)](#), or [RecognizeUtterance \(p. 710\)](#) operations.

Use a context to indicate to Amazon Lex V2 intents that should be used as follow-up intents. For example, if the active context is `order-fulfilled`, only intents that have `order-fulfilled` configured as a trigger are considered for follow up.

### Contents

#### **contextAttributes**

A list of contexts active for the request. A context can be activated when a previous intent is fulfilled, or by including the context in the request.

If you don't specify a list of contexts, Amazon Lex V2 will use the current list of contexts for the session. If you specify an empty list, all contexts for the session are cleared.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 10 items.

Key Length Constraints: Minimum length of 1. Maximum length of 100.

Value Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

#### **name**

The name of the context.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^([A-Za-z]\_?)+\$

Required: Yes

#### **timeToLive**

Indicates the number of turns or seconds that the context is active. Once the time to live expires, the context is no longer returned in a response.

Type: [ActiveContextTimeToLive \(p. 895\)](#) object

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## ActiveContextTimeToLive

Service: Amazon Lex Runtime V2

The time that a context is active. You can specify the time to live in seconds or in conversation turns.

### Contents

#### timeToLiveInSeconds

The number of seconds that the context is active. You can specify between 5 and 86400 seconds (24 hours).

Type: Integer

Valid Range: Minimum value of 5. Maximum value of 86400.

Required: Yes

#### turnsToLive

The number of turns that the context is active. You can specify up to 20 turns. Each request and response from the bot is a turn.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 20.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AudioInputEvent

Service: Amazon Lex Runtime V2

Represents a chunk of audio sent from the client application to Amazon Lex V2. The audio is all or part of an utterance from the user.

Amazon Lex V2 accumulates audio chunks until it recognizes a natural pause in speech before processing the input.

### Contents

#### audioChunk

An encoded stream of audio.

Type: Base64-encoded binary data object

Required: No

#### clientTimestampMillis

A timestamp set by the client of the date and time that the event was sent to Amazon Lex V2.

Type: Long

Required: No

#### contentType

The encoding used for the audio chunk. You must use 8 KHz PCM 16-bit mono-channel little-endian format. The value of the field should be:

`audio/lpcm; sample-rate=8000; sample-size-bits=16; channel-count=1; is-big-endian=false`

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

#### eventId

A unique identifier that your application assigns to the event. You can use this to identify events in logs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: `[0-9a-zA-Z._:-]+`

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

## AudioResponseEvent

Service: Amazon Lex Runtime V2

An event sent from Amazon Lex V2 to your client application containing audio to play to the user.

### Contents

#### **audioChunk**

A chunk of the audio to play.

Type: Base64-encoded binary data object

Required: No

#### **contentType**

The encoding of the audio chunk. This is the same as the encoding configure in the `contentType` field of the ConfigurationEvent.

Type: String

Length Constraints: Minimum length of 1.

Required: No

#### **eventId**

A unique identifier of the event sent by Amazon Lex V2. The identifier is in the form RESPONSE-N, where N is a number starting with one and incremented for each event sent by Amazon Lex V2 in the current session.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Button

Service: Amazon Lex Runtime V2

A button that appears on a response card show to the user.

### Contents

#### **text**

The text that is displayed on the button.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 50.

Required: Yes

#### **value**

The value returned to Amazon Lex V2 when a user chooses the button.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 50.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ConfidenceScore

Service: Amazon Lex Runtime V2

Provides a score that indicates the confidence that Amazon Lex V2 has that an intent is the one that satisfies the user's intent.

### Contents

#### score

A score that indicates how confident Amazon Lex V2 is that an intent satisfies the user's intent. Ranges between 0.00 and 1.00. Higher scores indicate higher confidence.

Type: Double

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ConfigurationEvent

Service: Amazon Lex Runtime V2

The initial event sent from the application to Amazon Lex V2 to configure the conversation, including session and request attributes and the response content type.

### Contents

#### clientTimestampMillis

A timestamp set by the client of the date and time that the event was sent to Amazon Lex V2.

Type: Long

Required: No

#### disablePlayback

Determines whether Amazon Lex V2 should send audio responses to the client application.

Set this field to false when the client is operating in a playback mode where audio responses are played to the user. If the client isn't operating in playback mode, such as a text chat application, set this to true so that Amazon Lex V2 doesn't wait for the prompt to finish playing on the client.

Type: Boolean

Required: No

#### eventId

A unique identifier that your application assigns to the event. You can use this to identify events in logs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### requestAttributes

Request-specific information passed between the client application and Amazon Lex V2.

The namespace x-amz-lex: is reserved for special attributes. Don't create any request attributes for prefix x-amz-lex:.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

#### responseContentType

The message that Amazon Lex V2 returns in the response can be either text or speech based on the responseContentType value.

- If the value is text/plain; charset=utf-8, Amazon Lex V2 returns text in the response.
- If the value begins with audio/, Amazon Lex V2 returns speech in the response. Amazon Lex V2 uses Amazon Polly to generate the speech using the configuration that you specified in the requestContentType parameter. For example, if you specify audio/mpeg as the value, Amazon Lex V2 returns speech in the MPEG format.

- If the value is audio/pcm, the speech returned is audio/pcm in 16-bit, little-endian format.
- The following are the accepted values:
  - audio/mpeg
  - audio/ogg
  - audio/pcm
  - audio/\* (defaults to mpeg)
  - text/plain; charset=utf-8

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

#### **sessionState**

The state of the user's session with Amazon Lex V2.

Type: [SessionState \(p. 923\)](#) object

Required: No

#### **welcomeMessages**

A list of messages to send to the user.

If you set the welcomeMessage field, you must also set the [DialogAction](#) structure's [type](#) field.

Type: Array of [Message \(p. 915\)](#) objects

Array Members: Maximum number of 10 items.

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DialogAction

Service: Amazon Lex Runtime V2

The next action that Amazon Lex V2 should take.

### Contents

#### slotElicitationStyle

Configures the slot to use spell-by-letter or spell-by-word style. When you use a style on a slot, users can spell out their input to make it clear to your bot.

- Spell by letter - "b" "o" "b"
- Spell by word - "b as in boy" "o as in oscar" "b as in boy"

For more information, see [Using spelling to enter slot values](#).

Type: String

Valid Values: Default | SpellByLetter | SpellByWord

Required: No

#### slotToElicit

The name of the slot that should be elicited from the user.

Type: String

Length Constraints: Minimum length of 1.

Required: No

#### subSlotToElicit

The name of the constituent sub slot of the composite slot specified in slotToElicit that should be elicited from the user.

Type: [ElicitSubSlot \(p. 907\)](#) object

Required: No

#### type

The next action that the bot should take in its interaction with the user. The possible values are:

- Close - Indicates that there will not be a response from the user. For example, the statement "Your order has been placed" does not require a response.
- ConfirmIntent - The next action is asking the user if the intent is complete and ready to be fulfilled. This is a yes/no question such as "Place the order?"
- Delegate - The next action is determined by Amazon Lex V2.
- ElicitIntent - The next action is to elicit an intent from the user.
- ElicitSlot - The next action is to elicit a slot value from the user.

Type: String

Valid Values: Close | ConfirmIntent | Delegate | ElicitIntent | ElicitSlot | None

Required: Yes

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DisconnectionEvent

Service: Amazon Lex Runtime V2

A notification from the client that it is disconnecting from Amazon Lex V2. Sending a DisconnectionEvent event is optional, but can help identify a conversation in logs.

### Contents

#### **clientTimestampMillis**

A timestamp set by the client of the date and time that the event was sent to Amazon Lex V2.

Type: Long

Required: No

#### **eventId**

A unique identifier that your application assigns to the event. You can use this to identify events in logs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DTMFInputEvent

Service: Amazon Lex Runtime V2

A DTMF character sent from the client application. DTMF characters are typically sent from a phone keypad to represent numbers. For example, you can have Amazon Lex V2 process a credit card number input from a phone.

### Contents

#### clientTimestampMillis

A timestamp set by the client of the date and time that the event was sent to Amazon Lex V2.

Type: Long

Required: No

#### eventId

A unique identifier that your application assigns to the event. You can use this to identify events in logs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### inputCharacter

The DTMF character that the user pressed. The allowed characters are A - D, 0 - 9, # and \*.

Type: String

Length Constraints: Fixed length of 1.

Pattern: ^[A-D0-9#\*]{1}\$

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ElicitSubSlot

Service: Amazon Lex Runtime V2

The specific constituent sub slot of the composite slot to elicit in dialog action.

### Contents

#### **name**

The name of the slot that should be elicited from the user.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

#### **subSlotToElicit**

The field is not supported.

Type: [ElicitSubSlot \(p. 907\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## HeartbeatEvent

Service: Amazon Lex Runtime V2

Event that Amazon Lex V2 sends to indicate that the stream is still open between the client application and Amazon Lex V2

### Contents

#### eventId

A unique identifier of the event sent by Amazon Lex V2. The identifier is in the form RESPONSE-N, where N is a number starting with one and incremented for each event sent by Amazon Lex V2 in the current session.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ImageResponseCard

Service: Amazon Lex Runtime V2

A card that is shown to the user by a messaging platform. You define the contents of the card, the card is displayed by the platform.

When you use a response card, the response from the user is constrained to the text associated with a button on the card.

### Contents

#### buttons

A list of buttons that should be displayed on the response card. The arrangement of the buttons is determined by the platform that displays the button.

Type: Array of [Button \(p. 899\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 5 items.

Required: No

#### imageUrl

The URL of an image to display on the response card. The image URL must be publicly available so that the platform displaying the response card has access to the image.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 250.

Required: No

#### subtitle

The subtitle to display on the response card. The format of the subtitle is determined by the platform displaying the response card.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 250.

Required: No

#### title

The title to display on the response card. The format of the title is determined by the platform displaying the response card.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 250.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Intent

Service: Amazon Lex Runtime V2

The current intent that Amazon Lex V2 is attempting to fulfill.

### Contents

#### confirmationState

Contains information about whether fulfillment of the intent has been confirmed.

Type: String

Valid Values: Confirmed | Denied | None

Required: No

#### name

The name of the intent.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

#### slots

A map of all of the slots for the intent. The name of the slot maps to the value of the slot. If a slot has not been filled, the value is null.

Type: String to [Slot \(p. 925\)](#) object map

Key Length Constraints: Minimum length of 1.

Required: No

#### state

Contains fulfillment information for the intent.

Type: String

Valid Values: Failed | Fulfilled | InProgress | ReadyForFulfillment | Waiting | FulfillmentInProgress

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntentResultEvent

Service: Amazon Lex Runtime V2

Contains the current state of the conversation between the client application and Amazon Lex V2.

### Contents

#### eventId

A unique identifier of the event sent by Amazon Lex V2. The identifier is in the form RESPONSE-N, where N is a number starting with one and incremented for each event sent by Amazon Lex V2 in the current session.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### inputMode

Indicates whether the input to the operation was text or speech.

Type: String

Valid Values: Text | Speech | DTMF

Required: No

#### interpretations

A list of intents that Amazon Lex V2 determined might satisfy the user's utterance.

Each interpretation includes the intent, a score that indicates how confident Amazon Lex V2 is that the interpretation is the correct one, and an optional sentiment response that indicates the sentiment expressed in the utterance.

Type: Array of [Interpretation \(p. 914\)](#) objects

Array Members: Maximum number of 5 items.

Required: No

#### requestAttributes

The attributes sent in the request.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

#### sessionId

The identifier of the session in use.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

**sessionState**

The state of the user's session with Amazon Lex V2.

Type: [SessionState \(p. 923\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Interpretation

Service: Amazon Lex Runtime V2

An intent that Amazon Lex V2 determined might satisfy the user's utterance. The intents are ordered by the confidence score.

### Contents

#### **intent**

A list of intents that might satisfy the user's utterance. The intents are ordered by the confidence score.

Type: [Intent \(p. 911\)](#) object

Required: No

#### **nluConfidence**

Determines the threshold where Amazon Lex V2 will insert the AMAZON.FallbackIntent, AMAZON.KendraSearchIntent, or both when returning alternative intents in a response. AMAZON.FallbackIntent and AMAZON.KendraSearchIntent are only inserted if they are configured for the bot.

Type: [ConfidenceScore \(p. 900\)](#) object

Required: No

#### **sentimentResponse**

The sentiment expressed in an utterance.

When the bot is configured to send utterances to Amazon Comprehend for sentiment analysis, this field contains the result of the analysis.

Type: [SentimentResponse \(p. 921\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Message

Service: Amazon Lex Runtime V2

Container for text that is returned to the customer..

### Contents

#### **content**

The text of the message.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

#### **contentType**

Indicates the type of response.

Type: String

Valid Values: CustomPayload | ImageResponseCard | PlainText | SSML

Required: Yes

#### **imageResponseCard**

A card that is shown to the user by a messaging platform. You define the contents of the card, the card is displayed by the platform.

When you use a response card, the response from the user is constrained to the text associated with a button on the card.

Type: [ImageResponseCard \(p. 909\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PlaybackCompletionEvent

Service: Amazon Lex Runtime V2

Event sent from the client application to Amazon Lex V2 to indicate that playback of audio is complete and that Amazon Lex V2 should start processing the user's input.

### Contents

#### clientTimestampMillis

A timestamp set by the client of the date and time that the event was sent to Amazon Lex V2.

Type: Long

Required: No

#### eventId

A unique identifier that your application assigns to the event. You can use this to identify events in logs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PlaybackInterruptionEvent

Service: Amazon Lex Runtime V2

Event sent from Amazon Lex V2 to indicate to the client application should stop playback of audio. For example, if the client is playing a prompt that asks for the user's telephone number, the user might start to say the phone number before the prompt is complete. Amazon Lex V2 sends this event to the client application to indicate that the user is responding and that Amazon Lex V2 is processing their input.

### Contents

#### causedByEventId

The identifier of the event that contained the audio, DTMF, or text that caused the interruption.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### eventId

A unique identifier of the event sent by Amazon Lex V2. The identifier is in the form RESPONSE-N, where N is a number starting with one and incremented for each event sent by Amazon Lex V2 in the current session.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### eventReason

Indicates the type of user input that Amazon Lex V2 detected.

Type: String

Valid Values: DTMF\_START\_DETECTED | TEXT\_DETECTED | VOICE\_START\_DETECTED

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## RuntimeHintDetails

Service: Amazon Lex Runtime V2

Provides an array of phrases that should be given preference when resolving values for a slot.

### Contents

#### runtimeHintValues

One or more strings that Amazon Lex V2 should look for in the input to the bot. Each phrase is given preference when deciding on slot values.

Type: Array of [RuntimeHintValue \(p. 920\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

#### subSlotHints

A map of constituent sub slot names inside a composite slot in the intent and the phrases that should be added for each sub slot. Inside each composite slot hints, this structure provides a mechanism to add granular sub slot phrases. Only sub slot hints are supported for composite slots. The intent name, composite slot name and the constituent sub slot names must exist.

Type: String to [RuntimeHintDetails \(p. 918\)](#) object map

Key Length Constraints: Minimum length of 1. Maximum length of 100.

Key Pattern: ^([0-9a-zA-Z][\_-]?)+\$

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## RuntimeHints

Service: Amazon Lex Runtime V2

You can provide Amazon Lex V2 with hints to the phrases that a customer is likely to use for a slot. When a slot with hints is resolved, the phrases in the runtime hints are preferred in the resolution. You can provide hints for a maximum of 100 intents. You can provide a maximum of 100 slots.

Before you can use runtime hints with an existing bot, you must first rebuild the bot.

For more information, see [Using runtime hints to improve recognition of slot values](#).

### Contents

#### slotHints

A list of the slots in the intent that should have runtime hints added, and the phrases that should be added for each slot.

The first level of the `slotHints` map is the name of the intent. The second level is the name of the slot within the intent. For more information, see [Using hints to improve accuracy](#).

The intent name and slot name must exist.

Type: String to string to [RuntimeHintDetails \(p. 918\)](#) object map map

Key Length Constraints: Minimum length of 1. Maximum length of 100.

Key Pattern: `^([0-9a-zA-Z][_-]?)+$`

Key Length Constraints: Minimum length of 1. Maximum length of 100.

Key Pattern: `^([0-9a-zA-Z][_-]?)+$`

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## RuntimeHintValue

Service: Amazon Lex Runtime V2

Provides the phrase that Amazon Lex V2 should look for in the user's input to the bot.

### Contents

#### phrase

The phrase that Amazon Lex V2 should look for in the user's input to the bot.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 140.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SentimentResponse

Service: Amazon Lex Runtime V2

Provides information about the sentiment expressed in a user's response in a conversation. Sentiments are determined using Amazon Comprehend. Sentiments are only returned if they are enabled for the bot.

For more information, see [Determine Sentiment](#) in the *Amazon Comprehend developer guide*.

### Contents

#### **sentiment**

The overall sentiment expressed in the user's response. This is the sentiment most likely expressed by the user based on the analysis by Amazon Comprehend.

Type: String

Valid Values: MIXED | NEGATIVE | NEUTRAL | POSITIVE

Required: No

#### **sentimentScore**

The individual sentiment responses for the utterance.

Type: [SentimentScore \(p. 922\)](#) object

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SentimentScore

Service: Amazon Lex Runtime V2

The individual sentiment responses for the utterance.

### Contents

#### **mixed**

The level of confidence that Amazon Comprehend has in the accuracy of its detection of the MIXED sentiment.

Type: Double

Required: No

#### **negative**

The level of confidence that Amazon Comprehend has in the accuracy of its detection of the NEGATIVE sentiment.

Type: Double

Required: No

#### **neutral**

The level of confidence that Amazon Comprehend has in the accuracy of its detection of the NEUTRAL sentiment.

Type: Double

Required: No

#### **positive**

The level of confidence that Amazon Comprehend has in the accuracy of its detection of the POSITIVE sentiment.

Type: Double

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SessionState

Service: Amazon Lex Runtime V2

The state of the user's session with Amazon Lex V2.

### Contents

#### activeContexts

One or more contexts that indicate to Amazon Lex V2 the context of a request. When a context is active, Amazon Lex V2 considers intents with the matching context as a trigger as the next intent in a session.

Type: Array of [ActiveContext \(p. 893\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 20 items.

Required: No

#### dialogAction

The next step that Amazon Lex V2 should take in the conversation with a user.

Type: [DialogAction \(p. 903\)](#) object

Required: No

#### intent

The active intent that Amazon Lex V2 is processing.

Type: [Intent \(p. 911\)](#) object

Required: No

#### originatingRequestId

A unique identifier for a specific request.

Type: String

Length Constraints: Minimum length of 1.

Required: No

#### runtimeHints

Hints for phrases that a customer is likely to use for a slot. Amazon Lex V2 uses the hints to help determine the correct value of a slot.

Type: [RuntimeHints \(p. 919\)](#) object

Required: No

#### sessionAttributes

Map of key/value pairs representing session-specific context information. It contains application information passed between Amazon Lex V2 and a client application.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Slot

Service: Amazon Lex Runtime V2

A value that Amazon Lex V2 uses to fulfill an intent.

### Contents

#### shape

When the shape value is `List`, it indicates that the `values` field contains a list of slot values. When the value is `Scalar`, it indicates that the `value` field contains a single value.

Type: String

Valid Values: `Scalar` | `List` | `Composite`

Required: No

#### subSlots

The constituent sub slots of a composite slot.

Type: String to [Slot \(p. 925\)](#) object map

Key Length Constraints: Minimum length of 1.

Required: No

#### value

The current value of the slot.

Type: [Value \(p. 934\)](#) object

Required: No

#### values

A list of one or more values that the user provided for the slot. For example, if a for a slot that elicits pizza toppings, the values might be "pepperoni" and "pineapple."

Type: Array of [Slot \(p. 925\)](#) objects

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## StartConversationRequestEventStream

Service: Amazon Lex Runtime V2

Represents a stream of events between your application and Amazon Lex V2.

### Contents

#### AudioInputEvent

Speech audio sent from your client application to Amazon Lex V2. Audio starts accumulating when Amazon Lex V2 identifies a voice and continues until a natural pause in the speech is found before processing.

Type: [AudioInputEvent \(p. 896\)](#) object

Required: No

#### ConfigurationEvent

Configuration information sent from your client application to Amazon Lex V2

Type: [ConfigurationEvent \(p. 901\)](#) object

Required: No

#### DisconnectionEvent

Event sent from the client application to indicate to Amazon Lex V2 that the conversation is over.

Type: [DisconnectionEvent \(p. 905\)](#) object

Required: No

#### DTMFInputEvent

DTMF information sent to Amazon Lex V2 by your application. Amazon Lex V2 accumulates the DTMF information from when the user sends the first character and ends

- when there's a pause longer than the value configured for the end timeout.
- when there's a digit that is the configured end character.
- when Amazon Lex V2 accumulates characters equal to the maximum DTMF character configuration.

Type: [DTMFInputEvent \(p. 906\)](#) object

Required: No

#### PlaybackCompletionEvent

Event sent from the client application to Amazon Lex V2 to indicate that it has finished playing audio and that Amazon Lex V2 should start listening for user input.

Type: [PlaybackCompletionEvent \(p. 916\)](#) object

Required: No

#### TextInputEvent

Text sent from your client application to Amazon Lex V2. Each TextInputEvent is processed individually.

Type: [TextInputEvent \(p. 931\)](#) object

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## StartConversationResponseEventStream

Service: Amazon Lex Runtime V2

Represents a stream of events between Amazon Lex V2 and your application.

### Contents

#### AccessDeniedException

Exception thrown when the credentials passed with the request are invalid or expired. Also thrown when the credentials in the request do not have permission to access the StartConversation operation.

Type: Exception

HTTP Status Code: 403

Required: No

#### AudioResponseEvent

An event sent from Amazon Lex V2 to your client application containing audio to play to the user.

Type: [AudioResponseEvent \(p. 898\)](#) object

Required: No

#### BadGatewayException

Type: Exception

HTTP Status Code: 502

Required: No

#### ConflictException

Exception thrown when two clients are using the same AWS account, Amazon Lex V2 bot, and session ID.

Type: Exception

HTTP Status Code: 409

Required: No

#### DependencyFailedException

Type: Exception

HTTP Status Code: 424

Required: No

#### HeartbeatEvent

Event that Amazon Lex V2 sends to indicate that the stream is still open between the client application and Amazon Lex V2

Type: [HeartbeatEvent \(p. 908\)](#) object

Required: No

#### IntentResultEvent

Event sent from Amazon Lex V2 to the client application containing the current state of the conversation between the user and Amazon Lex V2.

Type: [IntentResultEvent \(p. 912\)](#) object

Required: No

#### **InternalServerException**

An error occurred with Amazon Lex V2.

Type: Exception

HTTP Status Code: 500

Required: No

#### **PlaybackInterruptionEvent**

Event sent from Amazon Lex V2 to indicate to the client application should stop playback of audio. For example, if the client is playing a prompt that asks for the user's telephone number, the user might start to say the phone number before the prompt is complete. Amazon Lex V2 sends this event to the client application to indicate that the user is responding and that Amazon Lex V2 is processing their input.

Type: [PlaybackInterruptionEvent \(p. 917\)](#) object

Required: No

#### **ResourceNotFoundException**

Exception thrown if one of the input parameters points to a resource that does not exist. For example, if the bot ID specified does not exist.

Type: Exception

HTTP Status Code: 404

Required: No

#### **TextResponseEvent**

The event sent from Amazon Lex V2 to your application with text to present to the user.

Type: [TextResponseEvent \(p. 932\)](#) object

Required: No

#### **ThrottlingException**

Exception thrown when your application exceeds the maximum number of concurrent requests.

Type: Exception

HTTP Status Code: 429

Required: No

#### **TranscriptEvent**

Event sent from Amazon Lex V2 to your client application that contains a transcript of voice audio.

Type: [TranscriptEvent \(p. 933\)](#) object

Required: No

#### **ValidationException**

Exception thrown when one or more parameters could not be validated. The message contains the name of the field that isn't valid.

Type: Exception

HTTP Status Code: 400

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TextInputEvent

Service: Amazon Lex Runtime V2

The event sent from your client application to Amazon Lex V2 with text input from the user.

### Contents

#### **clientTimestampMillis**

A timestamp set by the client of the date and time that the event was sent to Amazon Lex V2.

Type: Long

Required: No

#### **eventId**

A unique identifier that your application assigns to the event. You can use this to identify events in logs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### **text**

The text from the user. Amazon Lex V2 processes this as a complete statement.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TextResponseEvent

Service: Amazon Lex Runtime V2

The event sent from Amazon Lex V2 to your application with text to present to the user.

### Contents

#### eventId

A unique identifier of the event sent by Amazon Lex V2. The identifier is in the form RESPONSE-N, where N is a number starting with one and incremented for each event sent by Amazon Lex V2 in the current session.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### messages

A list of messages to send to the user. Messages are ordered based on the order that you returned the messages from your Lambda function or the order that the messages are defined in the bot.

Type: Array of [Message \(p. 915\)](#) objects

Array Members: Maximum number of 10 items.

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TranscriptEvent

Service: Amazon Lex Runtime V2

Event sent from Amazon Lex V2 to your client application that contains a transcript of voice audio.

### Contents

#### eventId

A unique identifier of the event sent by Amazon Lex V2. The identifier is in the form RESPONSE-N, where N is a number starting with one and incremented for each event sent by Amazon Lex V2 in the current session.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 100.

Pattern: [0-9a-zA-Z.\_:-]+

Required: No

#### transcript

The transcript of the voice audio from the user.

Type: String

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Value

Service: Amazon Lex Runtime V2

The value of a slot.

### Contents

#### interpretedValue

The value that Amazon Lex V2 determines for the slot. The actual value depends on the setting of the value selection strategy for the bot. You can choose to use the value entered by the user, or you can have Amazon Lex V2 choose the first value in the `resolvedValues` list.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

#### originalValue

The text of the utterance from the user that was entered for the slot.

Type: String

Length Constraints: Minimum length of 1.

Required: No

#### resolvedValues

A list of additional values that have been recognized for the slot.

Type: Array of strings

Length Constraints: Minimum length of 1.

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see [Signature Version 4 Signing Process](#) in the *Amazon Web Services General Reference*.

### Action

The action to be performed.

Type: string

Required: Yes

#### Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

#### X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

#### X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4\_request"). The value is expressed in the following format: *access\_key/YYYYMMDD/region/service/aws4\_request*.

For more information, see [Task 2: Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

#### X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

#### X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

**X-Amz-Signature**

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

**X-Amz-SignedHeaders**

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see [Task 1: Create a Canonical Request For Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

## Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

**AccessDeniedException**

You do not have sufficient access to perform this action.

HTTP Status Code: 400

**IncompleteSignature**

The request signature does not conform to AWS standards.

HTTP Status Code: 400

**InternalFailure**

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

**InvalidAction**

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

**InvalidClientTokenId**

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

**InvalidParameterCombination**

Parameters that must not be used together were used together.

HTTP Status Code: 400

**InvalidParameterValue**

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

**InvalidQueryParameter**

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

**MalformedQueryString**

The query string contains a syntax error.

HTTP Status Code: 404

**MissingAction**

The request is missing an action or a required parameter.

HTTP Status Code: 400

**MissingAuthenticationToken**

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

**MissingParameter**

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

**NotAuthorized**

You do not have permission to perform this action.

HTTP Status Code: 400

**OptInRequired**

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

**RequestExpired**

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

**ServiceUnavailable**

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

**ThrottlingException**

The request was denied due to request throttling.

HTTP Status Code: 400

**ValidationException**

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400