
Actions on Google Client Library

:warning: Warning: Conversational Actions will be deprecated on June 13, 2023. For more information, see [Conversational Actions Sunset](#).

This client library makes it easy to create Actions for the Google Assistant and supports Dialogflow, Actions SDK (v2), and Smart Home fulfillment.

:warning: We now recommend using Actions Builder or the Actions SDK to develop, test, and deploy Conversational Actions. For Conversational Actions built using Actions Builder see the [@assistant/-conversation](#) library.

- [Client Library GitHub repo](#)
- [Client Library reference docs](#)
- [Actions on Google docs](#)
- [Actions on Google samples](#)

npm v3.0.0 build unknown

Setup Instructions

Install the library with either `npm install actions-on-google` or `yarn add actions-on-google` if you use yarn.

Developer Preview

To support features under Developer Preview, the library has a special `preview` branch which can be installed using the `@preview` tag.

This is installed with either `npm install actions-on-google@preview` or `yarn add actions-on-google@preview`.

The `preview` tag will be kept up to date with every new version of the library.

You can use the Developer Preview version to experience exciting new features that we're still testing to make sure we have the best developer experience, and help us providing feedback on the API design and feature set.

The APIs offered in Developer Preview have not matured to General Availability yet, which means: - **You can't publish Actions** that use features in Developer Preview. - The APIs are **potentially subject to backwards incompatible changes**.

Conversational Services

Dialogflow

```
1 // Import the appropriate service and chosen wrappers
2 const {
3   dialogflow,
4   Image,
5 } = require('actions-on-google')
6
7 // Create an app instance
8 const app = dialogflow()
9
10 // Register handlers for Dialogflow intents
11
12 app.intent('Default Welcome Intent', conv => {
13   conv.ask('Hi, how is it going?')
14   conv.ask('Here's a picture of a cat')
15   conv.ask(new Image({
16     url: 'https://developers.google.com/web/fundamentals/accessibility/
17           semantics-builtin/imgs/160204193356-01-cat-500.jpg',
18     alt: 'A cat',
19   }))
20 })
21
22 // Intent in Dialogflow called `Goodbye`
23 app.intent('Goodbye', conv => {
24   conv.close('See you later!')
25 })
26
27 app.intent('Default Fallback Intent', conv => {
28   conv.ask('I didn't understand. Can you tell me something else?')
```

Actions SDK

```
1 // Import the appropriate service and chosen wrappers
2 const {
3   actionssdk,
4   Image,
5 } = require('actions-on-google')
6
7 // Create an app instance
8 const app = actionssdk()
9
10 // Register handlers for Actions SDK intents
11
12 app.intent('actions.intent.MAIN', conv => {
13   conv.ask('Hi, how is it going?')
14   conv.ask('Here's a picture of a cat')
15   conv.ask(new Image({
16     url: 'https://developers.google.com/web/fundamentals/accessibility/
17           semantics-builtin/imgs/160204193356-01-cat-500.jpg',
```

```
17     alt: 'A cat',
18   }))
19 })
20
21 app.intent('actions.intent.TEXT', (conv, input) => {
22   if (input === 'bye' || input === 'goodbye') {
23     return conv.close('See you later!')
24   }
25   conv.ask(`I didn't understand. Can you tell me something else?`)
26 })
```

Notes about the code snippet

- `conv.ask/conv.close` can be called with any of the [Response](#) types.
- All [Helper](#) classes are of the [Response](#) type.

Dialogflow

- `app` is an instance of type [DialogflowApp](#).
- `app` accepts options of type [DialogflowOptions](#).
- `conv` is an instance of type [DialogflowConversation](#).

Actions SDK

- `app` is an instance of type [ActionsSdkApp](#).
- `app` accepts options of type [ActionsSdkOptions](#).
- `conv` is an instance of type [ActionsSdkConversation](#).

Smart Home

```
1 // Import the appropriate service
2 const { smarthome } = require('actions-on-google')
3
4 // Create an app instance
5 const app = smarthome()
6
7 // Register handlers for Smart Home intents
8
9 app.onExecute((body, headers) => {
10   return {
11     requestId: 'ff36...',
12     payload: {
13       // ...
14     },
15   }
```

```
15   }
16 })
17
18 app.onQuery((body, headers) => {
19   return {
20     requestId: 'ff36...',
21     payload: {
22       // ...
23     },
24   }
25 })
26
27 app.onSync((body, headers) => {
28   return {
29     requestId: 'ff36...',
30     payload: {
31       // ...
32     },
33   }
34 })
```

Notes about the code snippet

- `app` is an instance of type `SmartHomeApp`.
- `app` accepts options of type `SmartHomeOptions`.

Frameworks

Export or run for your appropriate framework:

Firebase Functions

```
1 const functions = require('firebase-functions')
2
3 // ... app code here
4
5 exports. fulfillment = functions.https.onRequest(app)
```

Dialogflow Inline Editor

```
1 const functions = require('firebase-functions')
2
3 // ... app code here
4
5 // name has to be `dialogflowFirebaseFulfillment`
6 exports.dialogflowFirebaseFulfillment = functions.https.onRequest(app)
```

Self Hosted Express Server

```
1 const express = require('express')
2 const bodyParser = require('body-parser')
3
4 // ... app code here
5
6 const expressApp = express().use(bodyParser.json())
7
8 expressApp.post('/fulfillment', app)
9
10 expressApp.listen(3000)
```

AWS Lambda API Gateway

```
1 // ... app code here
2
3 exports.fulfillment = app
```

Next Steps

Take a look at the docs and samples linked at the top to get to know the platform and supported functionalities.

Library Development Instructions

This library uses [yarn](#) to run commands. Install yarn using instructions from <https://yarnpkg.com/en/docs/install> or with npm: `npm i -g yarn`.

Install the library dependencies with [yarn](#). If you want to run any of the sample apps, follow the instructions in the sample README.

Functionality

Public interfaces, classes, functions, objects, and properties are labeled with the JSDoc `@public` tag and exported at the top level. Everything else that is not labeled `@public` and exported at the top level is considered internal and may be changed.

This library supports the following Services: * Dialogflow v1 and v2 * Actions SDK **v2 only** * Smart Home

Actions SDK

This library supports only Actions SDK fulfillment version 2.

To ensure that your fulfillment uses version 2, set the `"fulfillmentApiVersion": 2` property in your action package.

References & Issues

- Questions? Go to StackOverflow, Assistant Developer Community on Reddit or Support.
- For bugs, please report an issue on Github.
- Actions on Google Documentation
- Actions on Google Codelabs.
- Webhook Boilerplate Template for Actions on Google.

Make Contributions

Please read and follow the steps in the CONTRIBUTING.md.

License

See LICENSE.

Terms

Your use of this sample is subject to, and by using or downloading the sample files you agree to comply with, the Google APIs Terms of Service.