Overview

Actions on Google is a developer platform that lets you create software to extend the functionality of the Google Assistant (//assistant.google.com/), Google's virtual personal assistant, across more than 500 million devices, including smart speakers, phones, cars, TVs, headphones, watches, and more.

Users engage Google Assistant in conversation to get things done, like buying groceries or booking a ride (for a complete list of what's possible now, see the Actions directory (//assistant.google.com/explore/).) As a developer, you can use Actions on Google to easily create and manage delightful and effective conversational experiences between users and your own 3rd-party fulfillment service.

Actions for the Google Assistant

Unlike with traditional mobile and desktop apps, which use computer-centric paradigms, users interact with ***Actions*** for the Assistant through natural-sounding, back and forth conversation.

Key Terms:

- **Intent**: A goal or task that users want to do, such as ordering coffee or finding a piece of music. In Actions on Google, this is represented as a unique identifier and the corresponding user utterances that can trigger the intent.
- **Action**: An interaction you build for the Assistant that supports a specific intent and has a corresponding fulfillment that processes the intent.
- **Fulfillment**: A service, app, feed, conversation, or other logic that handles an intent and carries out the corresponding Action.

To start a conversation, the user needs to invoke your Action through the Assistant. Users say or type a phrase like *"Hey Google, talk to Google IO 18"*. This tells the Assistant the name of the Action to talk to.

From this point onwards, the user is talking to your Action and giving it input. This conversation continues as a two-way dialog until the user's request is fulfilled or the conversation is finished.

Note: We highly recommend you go through our Coversation design processes and best practices (https://developers.google.com/actions/design/) when building your Actions. This also gives you the best chance for getting your Actions approved when you submit them for production release. To learn more about designing conversational user interfaces, see the conversation design guide (https://developers.google.com/actions/design/).

How Actions work with the Assistant

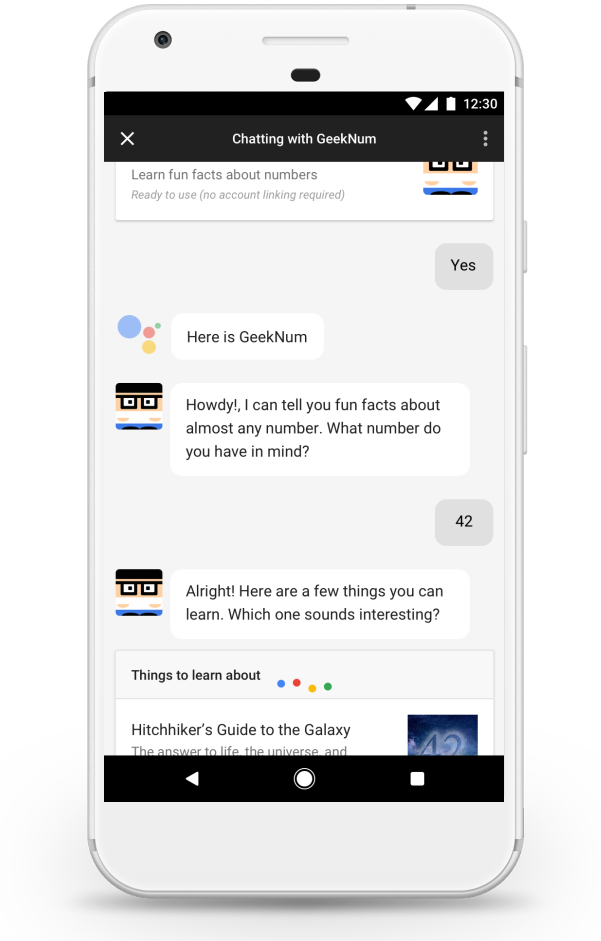

Figure 1: Example of using an Action on a mobile device

- When you build an Action for the Assistant, you design your conversations for a variety of surfaces, such as a voice-centric conversation for voice-activated speakers or a visual conversation on a surface that the Assistant supports. This lets users get things done quickly through either voice or visual affordances.
- Users can invoke your Actions from any of these surfaces:
    - Wear OS devices
    - Assistant-enabled headphones
    - Chromebooks
    - Android TV
    - Android phones and tablets
    - Smart displays and speakers
    - iPhones
- Your Actions run entirely in the cloud, even when users talk to them on their phone, smart home device, or watch.
- The user's device sends the user's utterance to the Assistant, which routes it to your fulfillment service via HTTP POST requests.
- Your fulfillment figures out a relevant response and sends that back to the Assistant, which ultimately returns it to the user.

Example: Learn about numbers with Geeknum

Figure 2: Example of using an Action on a Wear OS device

To understand how users interact with your Action, we'll use a hypothetical Action called 'Geeknum'. Users can invoke Geeknum through the Assistant on their phone or watch to learn fun facts about any number.

1. A user talks to the Assistant and requests an Action: *"I want to learn about numbers"*. The Assistant looks for the best Action to handle the user's intent and returns an Action named 'Geeknum'.
2. The Assistant asks the user if she wants to invoke Geeknum and she says *"Yes"*. Behind the scene, the Assistant sends a request to the Action's fulfillment by HTTP POST and receives a response that the Assistant renders to users.
3. The Assistant renders the response in its user interface and displays a welcome message to the user. The Assistant introduces Geeknum, then hands off the user to Geeknum. At this point, the conversation between the user and Geeknum begins.
4. During the conversation, the Assistant sends subsequent user input directly to the fulfillment and the fulfillment responds directly to the Assistant. This conversation continues until the fulfillment gathers the user input it needs to fulfill the user's intent.
5. The conversation ends when the user is done learning about numbers.

Last updated July 17, 2018.