

Custom events

Custom events are events that you define. You can invoke these events using either [fulfillment](/dialogflow/docs/fulfillment-overview) (/dialogflow/docs/fulfillment-overview) or [the API](/dialogflow/docs/api-overview) (/dialogflow/docs/api-overview). For example, you might set a timed alert during a conversation, which invokes an event at a certain time. This event could trigger an intent that alerts the end-user about something.

Invoke events with the API

The request for a [Sessions](/dialogflow/docs/reference/common-types#sessions) (/dialogflow/docs/reference/common-types#sessions) type `detectIntent` call contains a `queryInput.event` field, which is used to invoke events. The type of this field is a `EventInput`, and it contains fields for the event name, [optional parameters](/dialogflow/docs/events-overview#param) (/dialogflow/docs/events-overview#param), and the language code.

When you provide an event to a detect intent call, you do not provide other data, like an end-user expression. The call's sole purpose is to invoke an event and trigger an intent.

Invoke events with fulfillment

You can invoke events via [fulfillment](/dialogflow/docs/fulfillment-overview) (/dialogflow/docs/fulfillment-overview) by setting the `followupEventInput` field of the [WebhookResponse](/dialogflow/docs/reference/common-types#webhookresponse) (/dialogflow/docs/reference/common-types#webhookresponse). You can optionally set the `followupEventInput.parameters` field to provide [parameters](/dialogflow/docs/events-overview#param) (/dialogflow/docs/events-overview#param) to the intent.

For example:

```
followupEventInput": {
  name": "event-name",
  parameters": {
    "parameter-name-1": "parameter-value-1",
    "parameter-name-2": "parameter-value-2"
  },
  languageCode": "en-US"
```

When the `followupEventInput` parameter is set for a `WebhookResponse`, Dialogflow ignores the `fulfillmentText`, `fulfillmentMessages`, and `payload` fields. When Dialogflow receives a webhook response that includes an event, it immediately triggers the corresponding intent in which it was defined (</dialogflow/docs/events-overview#config>).

The following flow describes the steps:

1. The end-user types or speaks an expression.
2. Dialogflow matches the end-user expression to **Intent-1**, which is configured for fulfillment.
3. Dialogflow sends a webhook request to your server.
4. Your server responds with a webhook response that includes a followup event.
5. Instead of responding to the user for the **Intent-1** match, Dialogflow triggers **Intent-2**, which is configured for the event.
6. Dialogflow proceeds as though the end-user initiated the match for **Intent-2** and handles required parameters and fulfillment as dictated by the configuration of **Intent-2**.

Chaining intent matches with webhook response events

It is possible to chain intent matches with webhook response events. In the flow described above, **Intent-2** could also be configured for fulfillment. If your server's webhook response supplies another event, Dialogflow proceeds to match **Intent-3** (configured for the event) without responding to the end-user for **Intent-2**.

Dialogflow does not respond to the end-user until the chain has ended. When chaining intents in such a way, each webhook response must be sent within timeout restrictions (/dialogflow/docs/fulfillment-how#webhook_response).

You can chain *at most* 3 intent matches in this way.

[Previous](#)[← Platform events \(/dialogflow/docs/events-platform\)](#)[Next](#)[Fulfillment \(/dialogflow/docs/fulfillment-overview\) →](#)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-08-20 UTC.