

Detect intent with sentiment analysis

This feature is currently only available to users of a paid [edition](/dialogflow/docs/editions) (/dialogflow/docs/editions).

Sentiment analysis inspects user input and identifies the prevailing subjective opinion, especially to determine a user's attitude as positive, negative, or neutral. When making a detect intent request, you can specify that sentiment analysis be performed, and the response will contain sentiment analysis values.

The Natural Language API is used by Dialogflow to perform this analysis. For more information on that API and documentation on interpreting Dialogflow sentiment analysis results:

- [Sentiment Analysis](https://cloud.google.com/natural-language/docs/basics#sentiment_analysis) (https://cloud.google.com/natural-language/docs/basics#sentiment_analysis)
- [Interpreting sentiment analysis values](https://cloud.google.com/natural-language/docs/basics#interpreting_sentiment_analysis_values) (https://cloud.google.com/natural-language/docs/basics#interpreting_sentiment_analysis_values)

In cases where the sentiment analysis score is 0, the returned `sentimentAnalysisResult` field will be empty. For explicit 0 values in the response, see the `$outputDefaults` [system parameter](https://cloud.google.com/apis/docs/system-parameters#definitions) (https://cloud.google.com/apis/docs/system-parameters#definitions).

Supported languages

For a list of supported languages, see the **sentiment** column on the [languages](/dialogflow/docs/reference/language) (/dialogflow/docs/reference/language) page. If you request sentiment analysis for an unsupported language, your detect intent request does not fail, but the `QueryResult.diagnostic_info` field contains error information.

Before you begin

This feature is only applicable when using the API for [end-user interactions](/dialogflow/docs/api-overview) (/dialogflow/docs/api-overview). If you are using an [integration](/dialogflow/docs/integrations) (/dialogflow/docs/integrations), you can skip this guide.

You should do the following before reading this guide:

1. Read [Dialogflow basics](/dialogflow/docs/basics) (/dialogflow/docs/basics).
2. Perform [setup steps](/dialogflow/docs/quick/setup) (/dialogflow/docs/quick/setup).

Create an agent


If you have not already created an agent, create one now:

1. Go to the [Dialogflow Console](https://dialogflow.cloud.google.com) (https://dialogflow.cloud.google.com).
2. If requested, sign in to the Dialogflow Console. See [Dialogflow console overview](/dialogflow/docs/console) (/dialogflow/docs/console) for more information.
3. Click **Create Agent** in the left sidebar menu. (If you already have other agents, click the agent name, scroll to the bottom and click **Create new agent**.)
4. Enter your agent's name, default language, and default time zone.
5. If you have already created a project, enter that project. If you want to allow the Dialogflow Console to create the project, select **Create a new Google project**.
6. Click the **Create** button.

Import the example file to your agent

The steps in this guide make assumptions about your agent, so you need to [import](/dialogflow/docs/agents-settings#export) (/dialogflow/docs/agents-settings#export) an agent prepared for this guide. When importing, these steps use the *restore* option, which overwrites all agent settings, intents, and entities.

To import the file, follow these steps:

1. Download the [room-booking-agent.zip](/dialogflow/docs/data/room-booking-agent.zip) (/dialogflow/docs/data/room-booking-agent.zip) file.
2. Go to the [Dialogflow Console](https://dialogflow.cloud.google.com) (https://dialogflow.cloud.google.com).
3. Select your agent.
4. Click the settings  button next to the agent name.


5. Select the **Export and Import** tab.
6. Select **Restore From Zip** and follow instructions to restore the zip file that you downloaded.

Agent settings for sentiment analysis

You can trigger sentiment analysis per detect intent request, or you can configure your agent to always return sentiment analysis results.

Even if an agent is configured to use sentiment analysis, [Actions on Google](https://developers.google.com/actions/dialogflow/) (://developers.google.com/actions/dialogflow/) requests will not receive sentiment analysis results.

To enable sentiment analysis for all queries:

1. Go to the [Dialogflow Console](https://dialogflow.cloud.google.com) (https://dialogflow.cloud.google.com).
2. Select an agent.
3. Click the settings  button next to the agent name.
4. Select the **Advanced** tab.
5. Toggle **Enable sentiment analysis for the current query** on.

Use the Dialogflow simulator

You can interact with the agent and receive sentiment analysis results via the [Dialogflow simulator](https://dialogflow/docs/console#simulator) (/dialogflow/docs/console#simulator):

1. Type "Thank you for helping me."
2. See the **SENTIMENT** section at the bottom of the simulator. It should show a positive sentiment score.
3. Next, type "It didn't work at all." in the simulator.
4. See the **SENTIMENT** section at the bottom of the simulator. It should show a negative sentiment score.

When a very negative sentiment is returned, you may want to hand off unsatisfied users to live agents, or to get a better understanding of which intents lead to the highest customer sentiment. An example of a handoff implementation is found on [GitHub](https://github.com/dialogflow/agent-human-handoff-nodejs) (<https://github.com/dialogflow/agent-human-handoff-nodejs>)

Detect intent

[REST & CMD LINE](#) [Java \(#java\)](#) [Node.js \(#node.js\)](#) [Python \(#python\)](#)

Call the `detectIntent` method and provide the `sentimentAnalysisRequestConfig` field.

Before using any of the request data below, make the following replacements:

- ***project-id***: your GCP project ID

HTTP method and URL:

```
POST https://dialogflow.googleapis.com/v2/projects/project-id/agent/sessions/1
```

Request JSON body:

```
{
  "queryParams": {
    "sentimentAnalysisRequestConfig": {
      "analyzeQueryTextSentiment": true
    }
  },
  "queryInput": {
    "text": {
      "text": "please reserve an amazing meeting room for six people",
      "languageCode": "en-US"
    }
  }
}
```

To send your request, expand one of these options:

+ curl (Linux, macOS, or Cloud Shell)

Note: Ensure you have set the [GOOGLE_APPLICATION_CREDENTIALS](#) (/docs/authentication/production) environment variable to your service account private key file path.

Save the request body in a file called `request.json`, and execute the following command:

```
curl -X POST \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token
-H "Content-Type: application/json; charset=utf-8" \
-d @request.json \
https://dialogflow.googleapis.com/v2/projects/project-id/agent/sessions/123456
```

+ PowerShell (Windows)

Note: Ensure you have set the [GOOGLE_APPLICATION_CREDENTIALS](#) (/docs/authentication/production) environment variable to your service account private key file path.

Save the request body in a file called `request.json`, and execute the following command:

```
$cred = gcloud auth application-default print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
  -Method POST `
  -Headers $headers `
  -ContentType: "application/json; charset=utf-8" `
  -InFile request.json `
  -Uri "https://dialogflow.googleapis.com/v2/projects/project-id/agent/session
```

You should receive a JSON response similar to the following:

```
{
  "responseId": "747ee176-acc5-46be-8d9a-b7ef9c2b9199",
```

```
"queryResult": {
  "queryText": "please reserve an amazing meeting room for six people",
  "action": "room.reservation",
  "parameters": {
    "date": "",
    "duration": "",
    "guests": 6,
    "location": "",
    "time": ""
  },
  "fulfillmentText": "I can help with that. Where would you like to reserve
  ...
  "sentimentAnalysisResult": {
    "queryTextSentiment": {
      "score": 0.8,
      "magnitude": 0.8
    }
  }
}
```

Notice that the `sentimentAnalysisResult` field contains `score` and `magnitude` values.

[Previous](#)

← [Detect intent with audio output](#) (/dialogflow/docs/how/detect-intent-tts)

[Next](#)

[Long-running operations](#) (/dialogflow/docs/how/long-running-operations) →

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.