

Google Home Final Presentation

Mehmet Kardan, Hanna Köb, Mathias Meinschad, Daniel Linter

University of Innsbruck - SIT

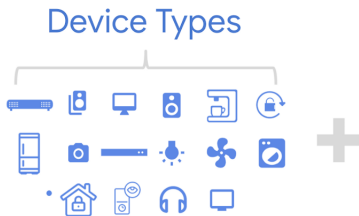
June 22, 2020

- 1 Overview
- 2 Execution Path
 - Intents
 - Entities
 - Fulfillment
- 3 Sparql Queries
- 4 Problems
- 5 Live Demo



- Founded by Google in 2016
- Development through Google's developer console and Dialogflow
- Creating skills pretty easy
- No programming skills required

Device Types & Traits



Traits

Attributes - SYNC

Defines configuration options for traits.

States - QUERY & EXECUTE

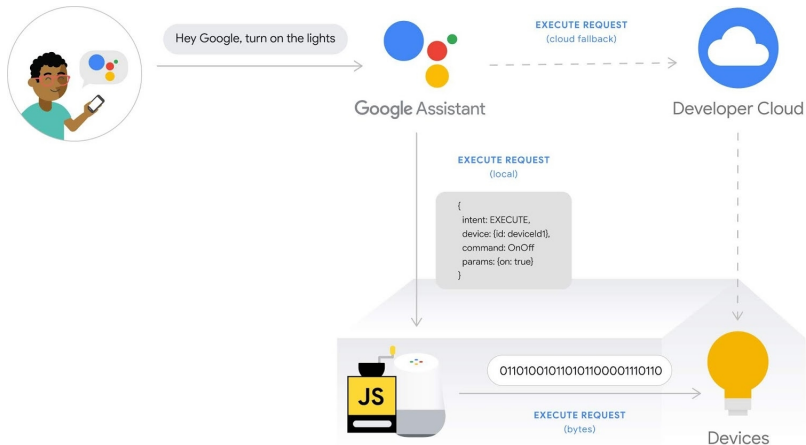
Defines the real-time state of the device.

Commands - EXECUTE

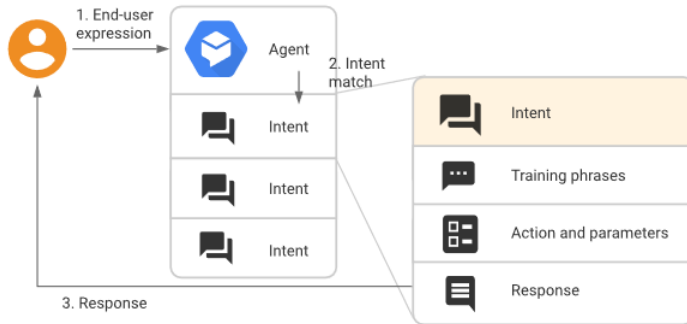
Used to change the state or perform a function on the device.

- Various device types (air purifier to yogurt maker)
- Capabilities of a device \Rightarrow traits

Execution Path



Intents



Intents cont'd

The screenshot displays the Google Dialogflow console interface for configuring an intent named 'annotation differences'. The left sidebar shows the project 'OcSeminarGroup3' and the 'Intents' tab is selected. The main area is divided into three sections: Training phrases, Action and parameters, and Responses.

Training phrases

Search training phrases

- 99 Add user expression
- 99 Explain the difference between a semi-automatic editing, manual editing, and automatic annotation
- 99 What is the difference between semi-automatic editing, manual editing, and automatic annotation

Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses

DEFAULT GOOGLE ASSISTANT +

Text Response

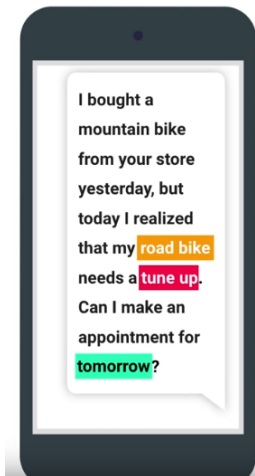
- 1 Response from knowledge graph

Try it now

Please use test console above to try a sentence.

See how it works in Google Assistant.

Entities



road bike
tune up
tomorrow
2017-11-09

System entities

@sys.time
@sys.date

Developer entities

@service-option

- Tune up
- Repair
- Tire change
- Upgrade

@bike-type

- Road bike
- Mountain bike
- Beach cruiser
- Racing bike
- Fixed gear bike
- Cross bike

Entities cont'd

The screenshot displays the Dialogflow console interface. On the left is a sidebar with navigation options: Intents, Entities (selected), Knowledge, Fulfillment, Integrations, Training, Validation, History, Analytics, Prebuilt Agents, Small Talk, Docs, and an Upgrade button. The main area is titled 'NLP_tasks' and contains a table of NLP tasks. Above the table are four checkboxes: 'Define synonyms' (checked), 'Regex entity' (unchecked), 'Allow automated expansion' (unchecked), and 'Fuzzy matching' (checked). The table lists various NLP tasks in two columns. The 'lemmatization' row is highlighted. Below the table is a '+ Add a row' link. On the right side of the console, there is a 'Try it now' button and a link to 'See how it works in Google Assistant'.

Dialogflow

DeSeminarGroup3

Intents

Entities

Knowledge

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Small Talk

Docs

Standard Free Upgrade

NLP_tasks **SAVE**

☒ Define synonyms ☐ Regex entity ☐ Allow automated expansion ☒ Fuzzy matching

tokenization	tokenization
stemming	stemming
lemmatization	lemmatization
sentence boundary disambiguation	sentence boundary disambiguation
named entity recognition	named entity recognition
part of speech tagging	part of speech tagging, part-of-speech tagging
chunking	chunking
syntactic parsing	syntactic parsing
semantic role labelling	semantic role labelling
coreference resolutions	coreference resolutions, co-reference resolutions, co reference resolutions

[Click here to edit entry](#)

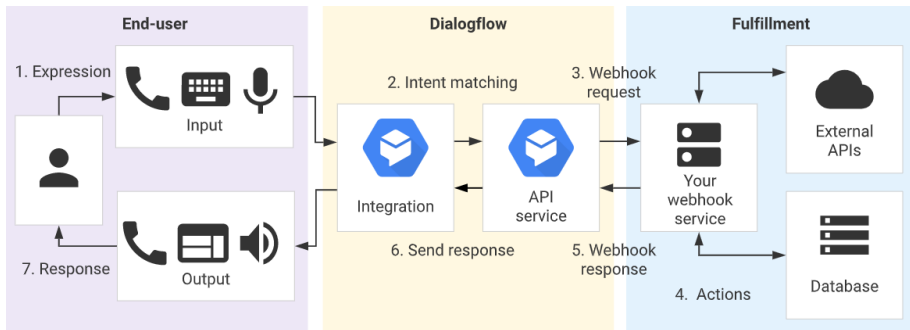
[+ Add a row](#)

Try it now

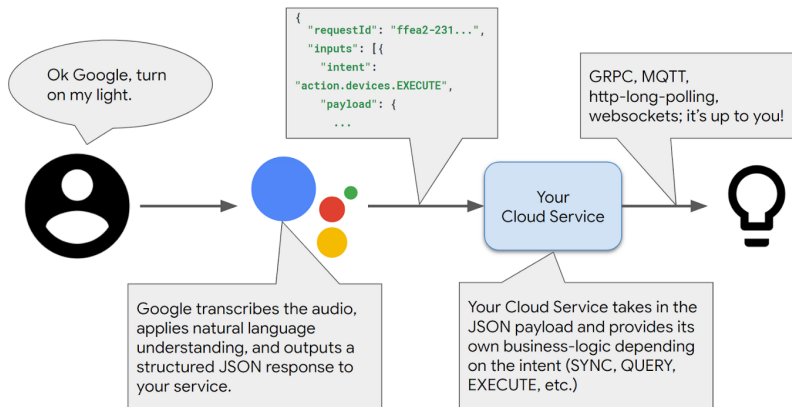
Please use test console above to try a sentence.

See how it works in Google Assistant.

Fulfillment



Communication



What Type Question

PREFIX schema: <http://schema.org/>

PREFIX kgs: <http://knowledgegraphbook.ai/schema/>

```
select ?description ?purpose where {  
  {  
    ?Concept schema:name ?name.  
    OPTIONAL { ?Concept schema:description ?description . }  
    OPTIONAL { ?Concept kgs:purpose ?purpose . }  
    filter (LCASE(?name) = LCASE("${parameter}"))  
  }  
  union  
  {  
    ?Concept schema:alternateName ?name.  
    OPTIONAL { ?Concept schema:description ?description . }  
    OPTIONAL { ?Concept kgs:purpose ?purpose . }  
    filter (LCASE(?name) = LCASE("${parameter}"))  
  }  
}
```

Difference Type Question

```
PREFIX schema: <http://schema.org/>
```

```
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>
```

```
select ?description where {  
  {  
    ?Concept schema:name ?name  
    OPTIONAL {?Concept kgbs:differsFrom ?relatesTo.}  
    OPTIONAL {?relatesTo schema:description ?description.}  
    filter (LCASE(?name) = LCASE("${first_parameter}") || LCASE(?name) = LCASE("${second_parameter}"))  
  }  
  UNION  
  {  
    ?Concept schema:alternateName ?name  
    OPTIONAL {?Concept kgbs:differsFrom ?relatesTo.}  
    OPTIONAL {?relatesTo schema:description ?description.}  
    filter (LCASE(?name) = LCASE("${first_parameter}") || LCASE(?name) = LCASE("${second_parameter}"))  
  }  
}
```

List Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

select ?description where {
  {
    ?Concept schema:name ?name
    OPTIONAL {?Concept skos:narrower ?specialization.}
    OPTIONAL {?specialization schema:name ?description.}
    filter (LCASE(?name) = LCASE("${parameter}")) .
  }
  union
  {
    ?Concept schema:alternateName ?name
    OPTIONAL {?Concept skos:narrower ?specialization.}
    OPTIONAL {?specialization schema:name ?description.}
    filter (LCASE(?name) = LCASE("${parameter}")) .
  }
}
```

Example Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
select ?description where {
  {
    ?Concept schema:name ?name.
    optional { ?Concept skos:example ?example . }
    optional { ?example schema:description ?description . }
    filter (LCASE(?name) = LCASE("${parameter}"))
  }
  UNION
  {
    ?Concept schema:alternateName ?name.|
    optional { ?Concept skos:example ?example . }
    optional { ?example schema:description ?description . }
    filter (LCASE(?name) = LCASE("${parameter}"))
  }
}
```

Step Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>

select ?description where {
  {
    ?Concept schema:name ?name .
    ?Concept schema:step: ?Object .
    OPTIONAL { ?Object schema:text ?description . }
    filter contains (LCASE(?name), LCASE("${parameter}")) .
  }
  UNION
  {
    ?Concept schema:alternateName ?name .
    ?Concept schema:step: ?Object .
    OPTIONAL { ?Object schema:text ?description . }
    filter contains (LCASE(?name), LCASE("${parameter}")) .
  }
}
```


Problems

Live Demo

Thank you for your
attention!