

Build a chatbot powered by Watson AI & Dashbot Analytics

IBM Code

Justina Nguyen, Dashbot Developer Advocate
Upkar Lidder, IBM Developer Advocate

A black and white photograph showing a group of approximately ten people working at computer desks in an office environment. Some individuals are looking at their screens, while others are interacting with each other or pointing at monitors. The scene conveys a sense of collaborative work and technology use.

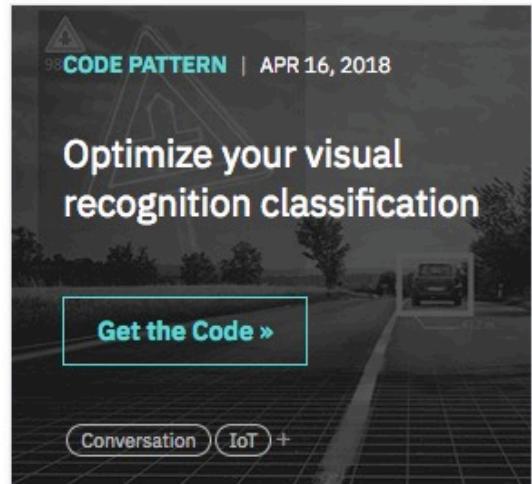
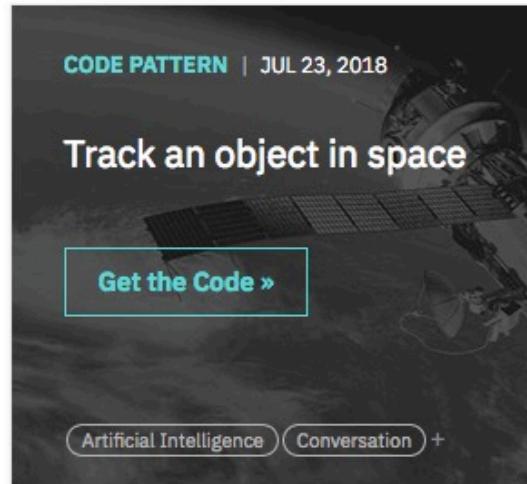
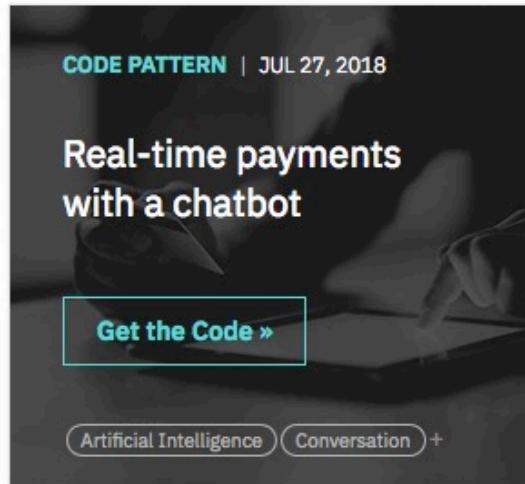
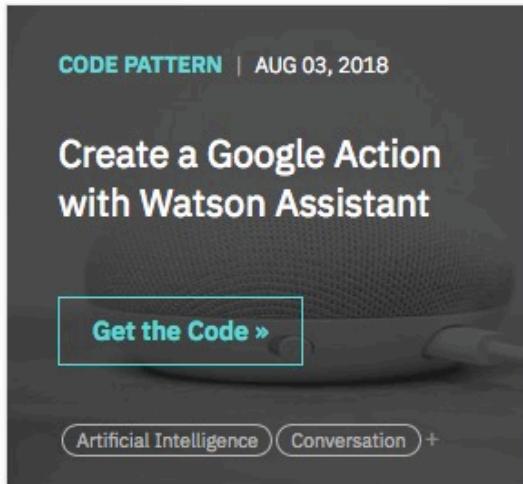
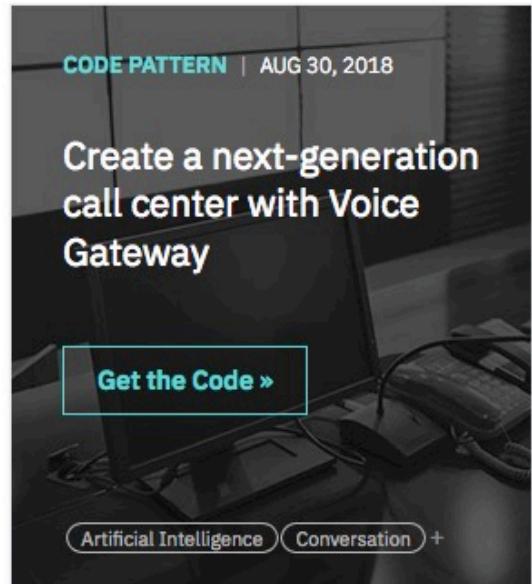
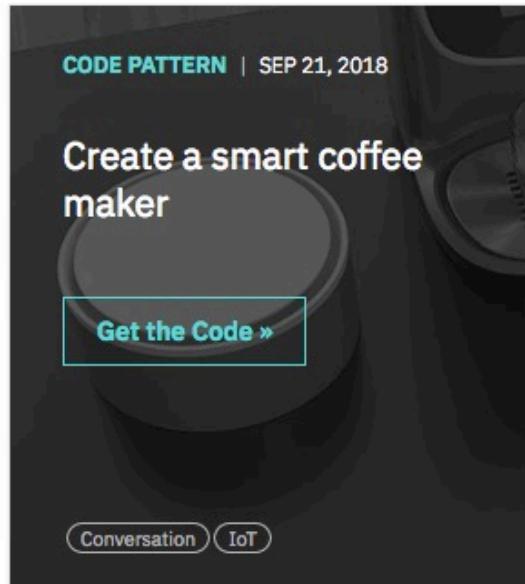
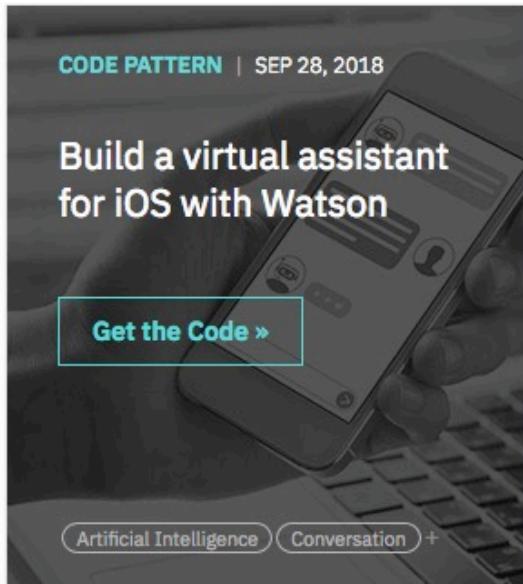
Wireless

IBM Signup: ibm.biz/ibm-sign-100318

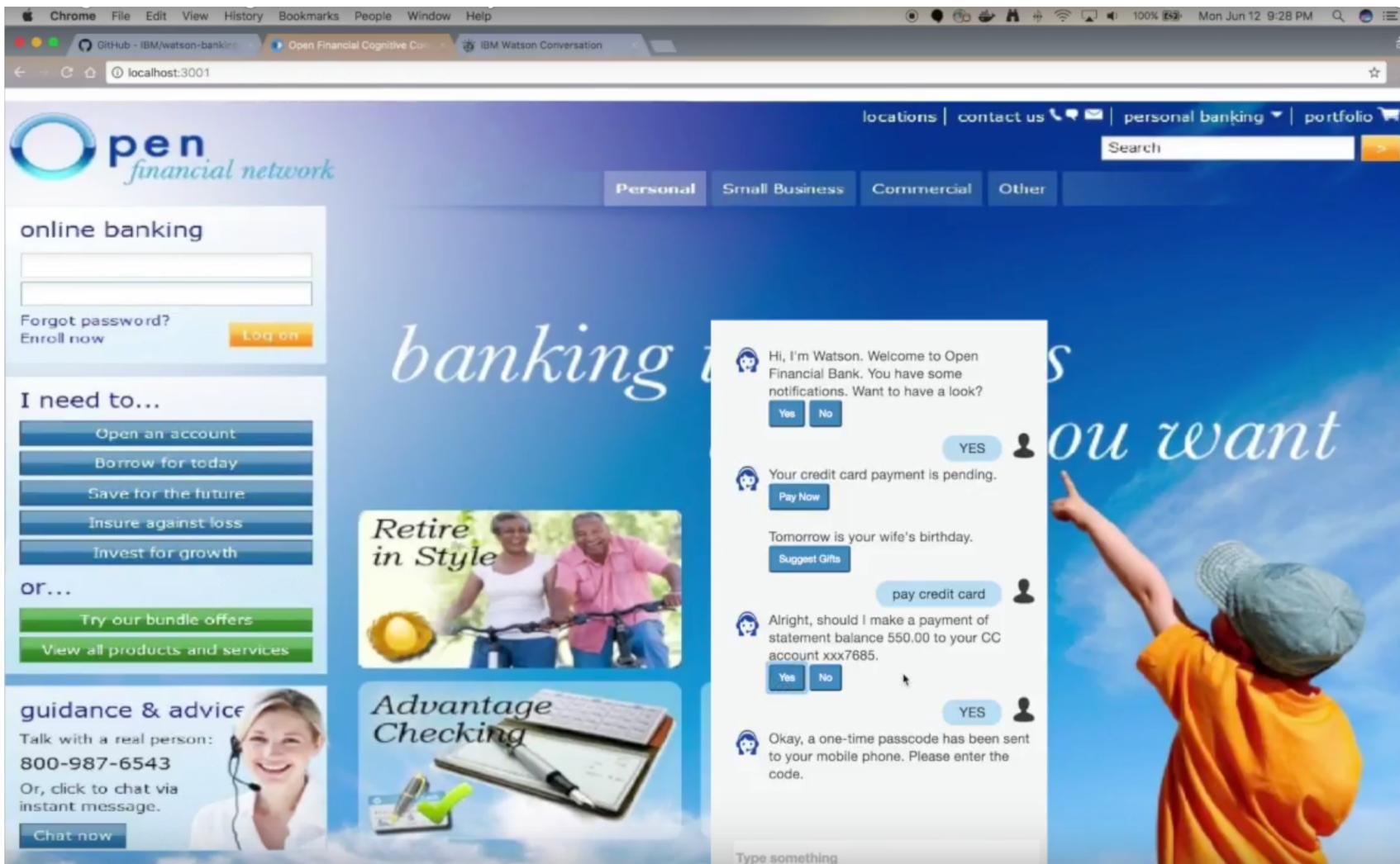
Dashbot Signup:

Github: ibm.biz/ibm-dashbot-100318

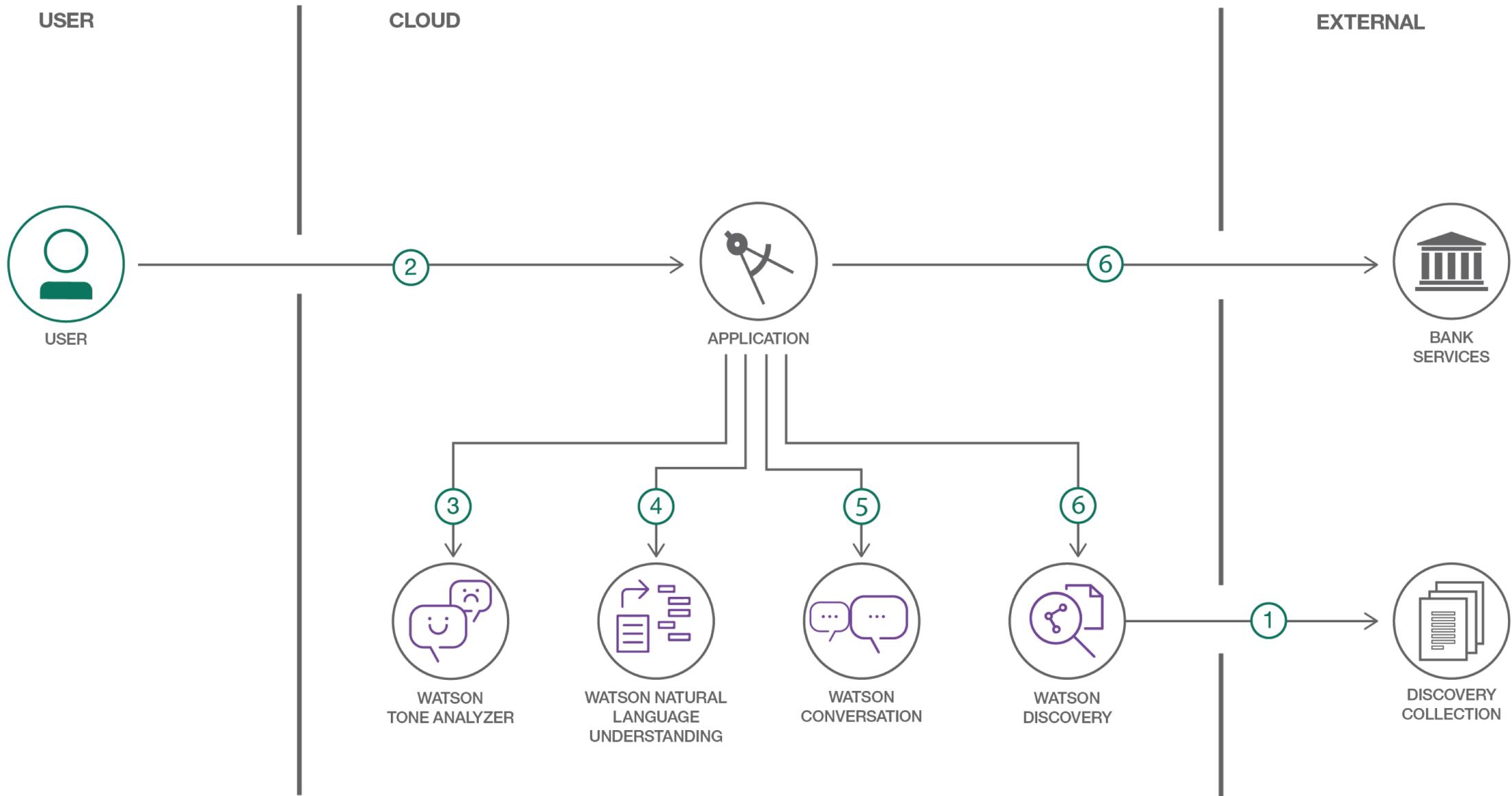
Code Patterns



Cognitive banking chatbot



Code Pattern - <https://developer.ibm.com/code/patterns/>



Watson Assistant

All Categories

Search

Filter

Infrastructure

Compute

Storage

Network

Security

Containers

VMware

Platform

Boilerplates

APIs

Application Services

Blockchain

Cloud Foundry Apps

Data & Analytics

DevOps

Finance

Functions

Integrate

Internet of Things

Mobile

Security

Storage

Watson

Build cognitive apps that help enhance, scale, and accelerate human expertise.



Watson Assistant (formerly Conversation)

Add a natural language interface to your application to automate interactions with your end users. Common

Lite

IBM



Discovery

Add a cognitive search and content analytics engine to applications.

Lite

IBM



Knowledge Catalog

Discover, catalog, and securely share enterprise data.

Lite

IBM



Knowledge Studio

Build custom models to teach Watson the language of your domain.

Lite

IBM



Language Translator

Translate text from one language to another, adapt translation models to your custom domain.

Lite

IBM



Machine Learning

IBM Watson Machine Learning - make smarter decisions, solve tough problems, and improve user outcomes.

Lite

IBM



Natural Language Classifier

Natural Language Classifier performs natural language classification on question texts. A user would be able to:

IBM



Natural Language Understanding

Analyze text to extract meta-data from content such as concepts, entities, emotion, relations, sentiment and more.

Lite

IBM



Speech to Text

Low-latency, streaming transcription

Lite

IBM



Text to Speech

Synthesizes natural-sounding speech from text.

Lite

IBM



Tone Analyzer

Tone Analyzer uses linguistic analysis to detect three types of tones from communications: emotion, social, and

Lite

IBM



Visual Recognition

Find meaning in visual content! Analyze images for scenes, objects, faces, and other content. Choose a default mo

Lite

IBM



Watson Studio

Embed AI and machine learning into your business. Create custom models using your own data.

Lite

IBM

FEEDBACK

Workshop Prep

<http://ibm.biz/ibm-dashbot-100318>

Workspace

Collection of artifacts – Intents,
Entities and Dialogs

Build

Deploy

Improve / Metrics

IBM Watson Assistant

The screenshot shows the IBM Watson Assistant interface. On the left is a vertical sidebar with four icons: wrenches (Intents), a circular arrow (Entities), a line graph (Dialog), and a grid (Content Catalog). The main area has a header with the title 'Workspaces / SampleWorkspace / Build' and tabs for 'Intents', 'Entities', 'Dialog', and 'Content Catalog'. Below the tabs is a large circular icon containing a network graph symbol. The text 'No intents yet.' is displayed. A descriptive paragraph explains that an intent is the goal or purpose of the user's input. It also provides links to 'Browse available intents' via the Content Catalog and three buttons: 'Add intent' (blue), 'Browse content catalog' (grey), and 'Import intents' (grey).

[Workspaces](#) / SampleWorkspace / Build

Intents Entities Dialog Content Catalog

No intents yet.

An intent is the goal or purpose of the user's input. Adding examples to intents helps your virtual assistant understand different ways in which people would say them.

Browse available intents to get started using the Content Catalog

Add intent

Browse content catalog

Import intents

Intents

An *intent* is the goal or purpose of the user's input. Adding examples to intents helps your virtual assistant understand different ways in which people would say them.

Provide at least five examples for each intent.

← #turn_on

Intent name
#turn_on

Description
Add a description to this intent

Add user examples
Add user examples to this intent

Add example

User examples (693) ▾

80's music

activate the jams

activate the music

air condition on

Air conditioner on

air on

Air on please.

also turn on the lights

Content Catalog

Intents Entities Dialog Content Catalog

Get started faster by adding existing intents from the content catalog. These intents are trained on common questions that users may ask.

Get started faster by adding existing intents from the content catalog. These intents are trained on common questions that users may ask.

Category	Description	Intents	
Banking	Basic transactions for a banking use case.	13	+ Add to workspace
Bot Control	Functions that allow navigation within a conversation.	9	+ Add to workspace
Customer Care	Understand and assist customers with information about themselves and your business.	18	+ Add to workspace
eCommerce	Payment, billing, and basic management tasks for orders.	14	+ Add to workspace
General	General conversation topics most users ask.	10	+ Add to workspace
Insurance	Issues related to insurance policies and claims.	12	+ Add to workspace
Telco	Questions and issues related to a user's telephony service, device, and plan.	21	+ Add to workspace
Utilities	Help a user with utility emergencies and their utility service.	10	+ Add to workspace

Entities

An *entity* is a portion of the user's input that you can use to provide a different response to a particular intent. Adding values and synonyms to entities helps your virtual assistant learn and understand important details that your users mention.

Entity name
@appliance

Value name
Enter value

Synonyms ▾

Add synonym... +

Add value

	Entity values (7) ▾	Type	
<input type="checkbox"/>	AC	Synonyms	ac, acs, air conditioner, air conditioners, air con
<input type="checkbox"/>	fan	Synonyms	cooler, fans
<input type="checkbox"/>	heater	Synonyms	heat, heating, heat vent, heat vents, hotter
<input type="checkbox"/>	lights	Synonyms	bulb, bulbs, foglight, fog light, foglights, fog lig
<input type="checkbox"/>	music	Synonyms	itune, itunes, music station, music stations, ra
<input type="checkbox"/>	volume	Synonyms	sound, speaker, speakers, vol
<input type="checkbox"/>	wipers	Synonyms	windscreen wiper, windscreen wipers, windsh

System Entities

System entities are common entities created by IBM that could be used across any use case. They are ready to use as soon as you add them.

Intents **Entities** Dialog Content Catalog

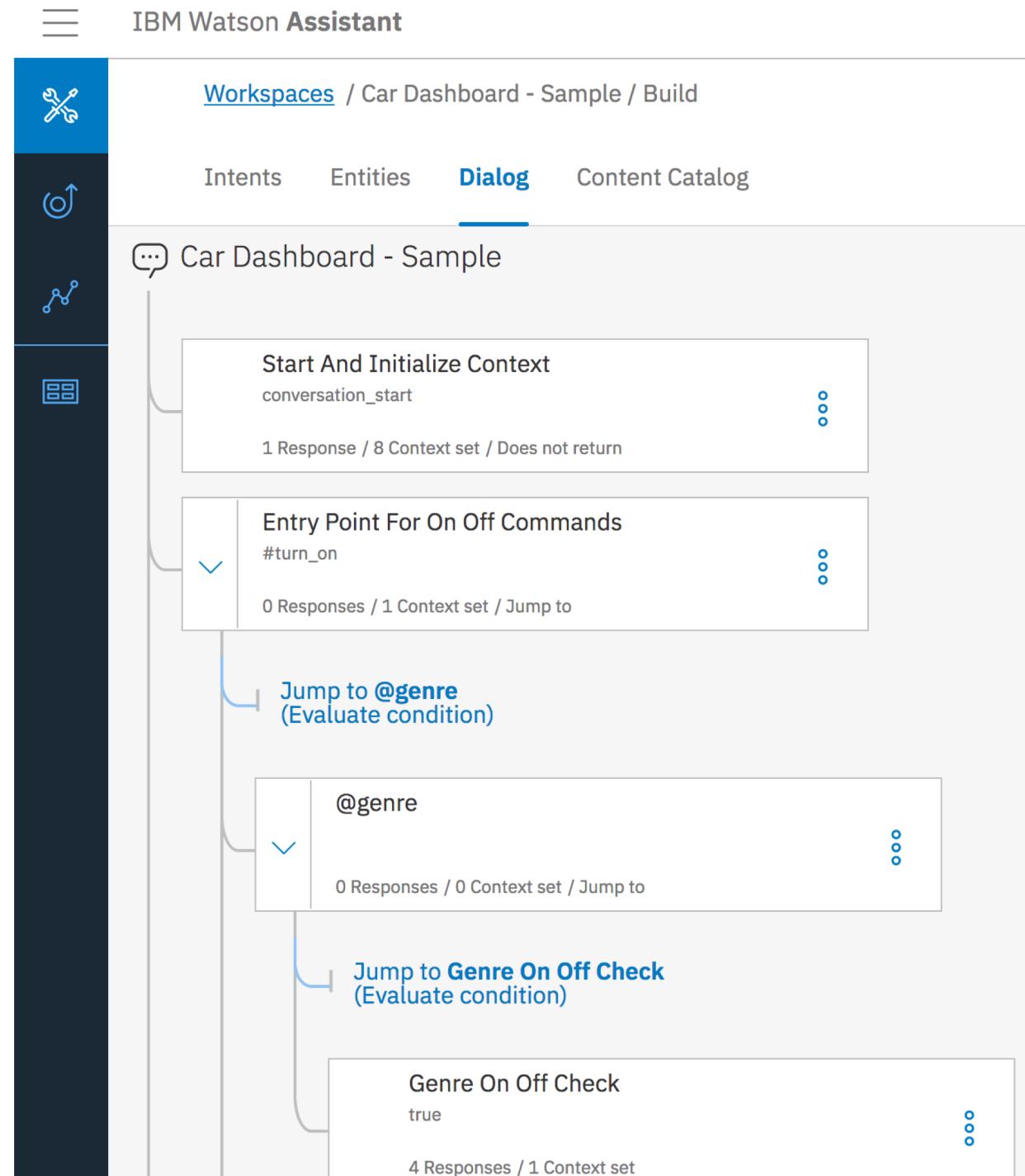
My entities **System entities**

- > **@sys-currency** Extracts currency values from user examples including the amount and the unit. (20 cents) Off
- > **@sys-date** Extracts date mentions (Friday) Off
- > **@sys-location BETA** The @sys-location system entity extracts place names (country, state/province, city, town, etc.) from the user's input. (Boston) Off
- > **@sys-number** Extracts numbers mentioned from user examples as digits or written as numbers. (21) Off
- > **@sys-percentage** Extracts amounts from user examples including the number and the % sign. (15%) Off
- > **@sys-person BETA** The @sys-person system entity extracts names from the user's input. (Anna) Off
- > **@sys-time** Extracts time mentions (at 10) Off

Dialog

A dialog uses intents, entities, and context from your application to define a response to each user's input.

Creating a dialog defines how your virtual assistant will respond to what your users are saying.

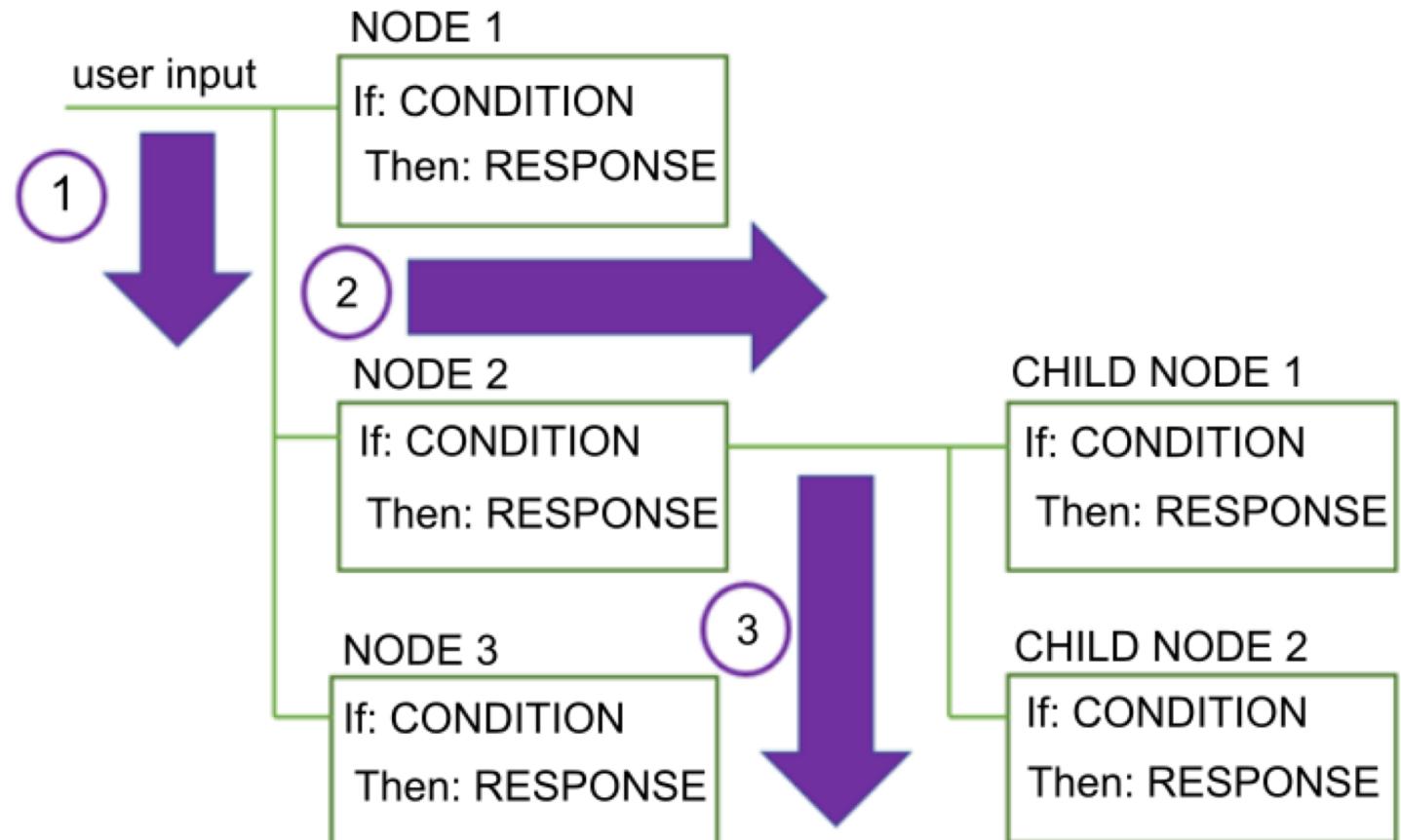


Dialog

Dialog is processed by the service from the first node in the tree to the last.

If the service finds a condition that is met, it triggers that node.

It then moves along the triggered node to check the user input against any child node conditions.



Slots

Add slots to a dialog node to gather multiple pieces of information from a user within that node.

Slots collect information at the users' pace. Details the user provides upfront are saved, and the service asks only for the details they do not.

Book Reservation

Then check for:

Check for	Save it as	If not present, ask
1 @cuisine	\$cuisine	What type of cuisine would you like?
2 @sys-date	\$date	What day would you like to eat?
3 @sys-time	\$time	What time would you like to eat?
4 @sys-number	\$number	How many people will be eating?

 Add slot

Digressions

Use digressions to handle interruptions and off topic short conversations.

Digressions allow the user to change topics or ask for more information and allows them to come back to the original intent.

Example:

Make an appointment.

I need to talk to an agent.

What times are you open?

The screenshot shows the IBM Watson Assistant interface with the 'Dialog' tab selected. A context node named 'Make an appointment' is highlighted. A modal window titled 'Customize "Make an appointment"' is open, with the 'Digressions' tab selected. The modal displays two sections: 'Digressions can go away from this node' and 'Digressions can come into this node'. Both sections have toggle switches set to 'on'. The 'Allow digressions away while slot filling' section includes a description: 'Users can divert the conversation away from this node in the middle of processing slots.' The 'Only digress from slots to nodes that allow returns' checkbox is unchecked. The 'Allow digressions into this node' section includes a description: 'Users can digress to this node from other dialog flows.' The 'Return after digression' checkbox is unchecked. At the bottom of the modal are 'Cancel' and 'Apply' buttons.

Handlers

Use handlers to

- Help the user respond to a slot.
- Configure individual slots.

IBM Watson Assistant
Workspaces / Customer Service - Sample / Dialog

Intents Entities Dialog Content

Opening welcome
1 Response / 1 Context set / Does not return

Hours of Operation #Customer_Care_Store_Hours
5 Responses / 0 Context set / Returns

Directions and location #Customer_Care_Store_Location
3 Responses / 0 Context set / Skip user input

Skip user input. The first child node

Provide location true
1 Response / 0 Context set

Make an appointment #Customer_Care_Appointments
3 Responses / 7 Context set / 5 Slots / Does not return

Transfer to agent #General_Connect_to_Agent
1 Response / 0 Context set / Does not return

Manage handlers for "Make an appointment"

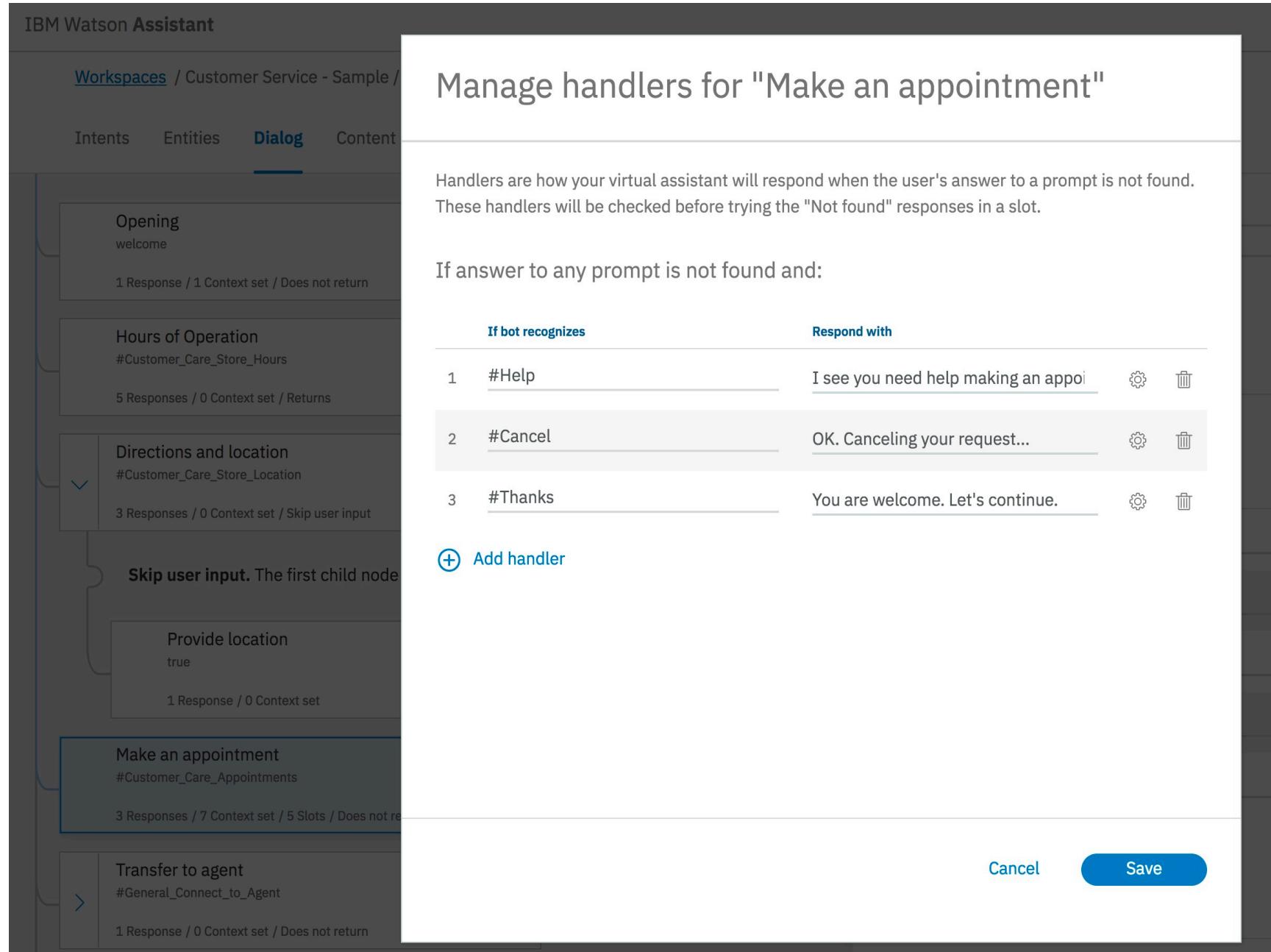
Handlers are how your virtual assistant will respond when the user's answer to a prompt is not found. These handlers will be checked before trying the "Not found" responses in a slot.

If answer to any prompt is not found and:

If bot recognizes	Respond with
1 #Help	I see you need help making an appoi  
2 #Cancel	OK. Canceling your request...  
3 #Thanks	You are welcome. Let's continue.  

+ Add handler

Cancel Save



Some tips before you start

- Start on paper and have a conversation with a friend (***discover intents, entities and create dialog***).
- Don't repeat yourself too many times - unless the information is absolutely necessary to meet the goal or help the end user (***slots***).
- Keep it natural - let the users digress and come back to finish the current conversation (***digressions***).
- Give the users a way out ! (***handlers***)
- Remember details about the users (***context***)
- Forget details when the user starts again (***empty out context***)
- Confirm details in a natural way “I have booked a table for two at 7pm tonight”.
- Enable Fuzzy logic – beta feature

Workshop Prep

<http://ibm.biz/ibm-dashbot-100318>

IBM Code SF – Upcoming Events

3
OCT

Wednesday, October 3, 2018, 6:00 PM

Hands-on chatbot workshop with Watson AI and Dashbot Analytics

31
OCT

Wednesday, October 31, 2018, 9:30 AM

Online Meetup: Deploy your bot to Slack and other social...

2
OCT

Tuesday, October 2, 2018, 6:00 PM

Managing and Securing Containers with Istio & Fun Tric...

7
NOV

Wednesday, November 7, 2018, 9:30 AM

Online Meetup: Continuously improve your Watson Bot using...

17
OCT

Wednesday, October 17, 2018, 9:30 AM

Online Meetup: Anatomy of a chatbot and building a...

25
OCT

Thursday, October 25, 2018, 9:30 AM

Online Meetup: Empower your chatbot and add a personality

18
OCT

Thursday, October 18, 2018, 6:00 PM

Securing Kubernetes with Kritis and Istio, a joint IBM/Google...

Open Data Science Conference (50% Discount for IBM)



The banner features a dark background with a molecular or network graph pattern in blue and grey. On the left, the hashtag #ODSC is displayed above a cluster of molecular structures. In the center, the text "ODSC West • San Francisco" is written in large white letters, followed by "2018" in a slightly smaller font. Below this, the dates "October 31st – November 3rd" and the location "• Hyatt Regency" are listed. At the bottom, there are two teal-colored buttons: one labeled "2018 PRELIMINARY SCHEDULE" and another labeled "REGISTER FOR 20% OFF".

Open Data Science Conference

ODSC EAST | EUROPE | WEST | INDIA | UKRAINE | Blog | Careers | Newsletter | Job Board | Partners | About | Home

#ODSC

ODSC West • San Francisco

2018

October 31st – November 3rd • Hyatt Regency

2018 PRELIMINARY SCHEDULE

REGISTER FOR 20% OFF

THANK YOU

Justina Nguyen, Dashbot Developer Advocate
Upkar Lidder, IBM Developer Advocate

Watson SDK

This organization Search Pull requests Issues Marketplace Explore

@watson-dev IBM Watson APIs

A collection of REST APIs and SDKs that use cognitive computing to solve complex problems.

📍 USA 🌐 https://www.ibm.com/watson/developer/

Repositories 84 People 30

Pinned repositories

node-sdk
Node.js library to access IBM Watson services.
TypeScript ★ 1k 516

python-sdk
Client library to use the IBM Watson services in Python and available in pip as watson-developer-cloud
Python ★ 849 509

swift-sdk
The Watson Swift SDK enables developers to quickly add Watson Cognitive Computing services to their Swift applications.
Swift ★ 724 186

java-sdk
Java SDK to use the IBM Watson services.
Java ★ 474 456

unity-sdk
Unity SDK to use the IBM Watson services.
C# ★ 339 135

dotnet-standard-sdk
.NET Standard library to access Watson Services.
C# ★ 79 67

JSON Response



Watson understands

Intent

Entity

User Input

Response

```
1  {
2      "intents": [
3          {
4              "intent": "turn_on",
5              "confidence": 0.9951137542724611
6          }
7      ],
8      "entities": [
9          {
10             "entity": "appliance",
11             "location": [
12                 12,
13                 22
14             ],
15             "value": "lights",
16             "confidence": 1
17         }
18     ],
19     "input": {
20         "text": "turn on the headlights"
21     },
22     "output": {
23         "text": [
24             "I'll turn on the lights for you."
25         ]
26     }
27 }
```

BotKit Middleware

The screenshot shows the GitHub repository page for 'watson-developer-cloud / botkit-middleware'. The repository has 106 commits, 2 branches, 6 releases, and 9 contributors. The latest commit was made 14 days ago. The repository description states: "A middleware to connect Watson Conversation Service to different chat channels using Botkit".

A middlewarer to connect Watson Conversation Service to different chat channels using Botkit
<https://www.npmjs.com/package/botkit-...>

Branch: master	New pull request	Create new file	Upload files	Find file	Clone or download
germanattanasio committed on GitHub Merge pull request #82 from Naktibalda/patch-2	...	Latest commit cd79ce4 14 days ago			
examples	Merge pull request #77 from Naktibalda/patch-1	28 days ago			
lib/middleware	updateContext ignores error of storage.users.get	a month ago			
test	updateContext ignores error of storage.users.get	a month ago			
.gitignore	Initial commit 🚀	11 months ago			
.npmignore	ignore the test folder	10 months ago			
.travis.yml	add travis file	10 months ago			
CHANGELOG.md	updateContext ignores error of storage.users.get	a month ago			
LICENSE	rename license file	10 months ago			
README.md	[Readme] Fixed typo in updateContext example	14 days ago			
package-lock.json	1.4.2	28 days ago			
package.json	1.4.2	28 days ago			

IBM Bot Asset Exchange

IBM Code

Code

Content

Community

Open Source

Bot Asset Exchange

Your community-driven chatbot development hub.

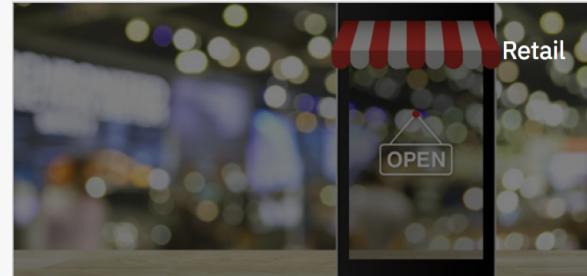
Discover, configure, deploy, and be rewarded for bots built with Watson Assistant. Join the revolution--build a chatbot today.

[Get a Bot](#)[Upload a Bot](#)[Learn More](#)

Featured bots



Shrink - The Emotion Bot

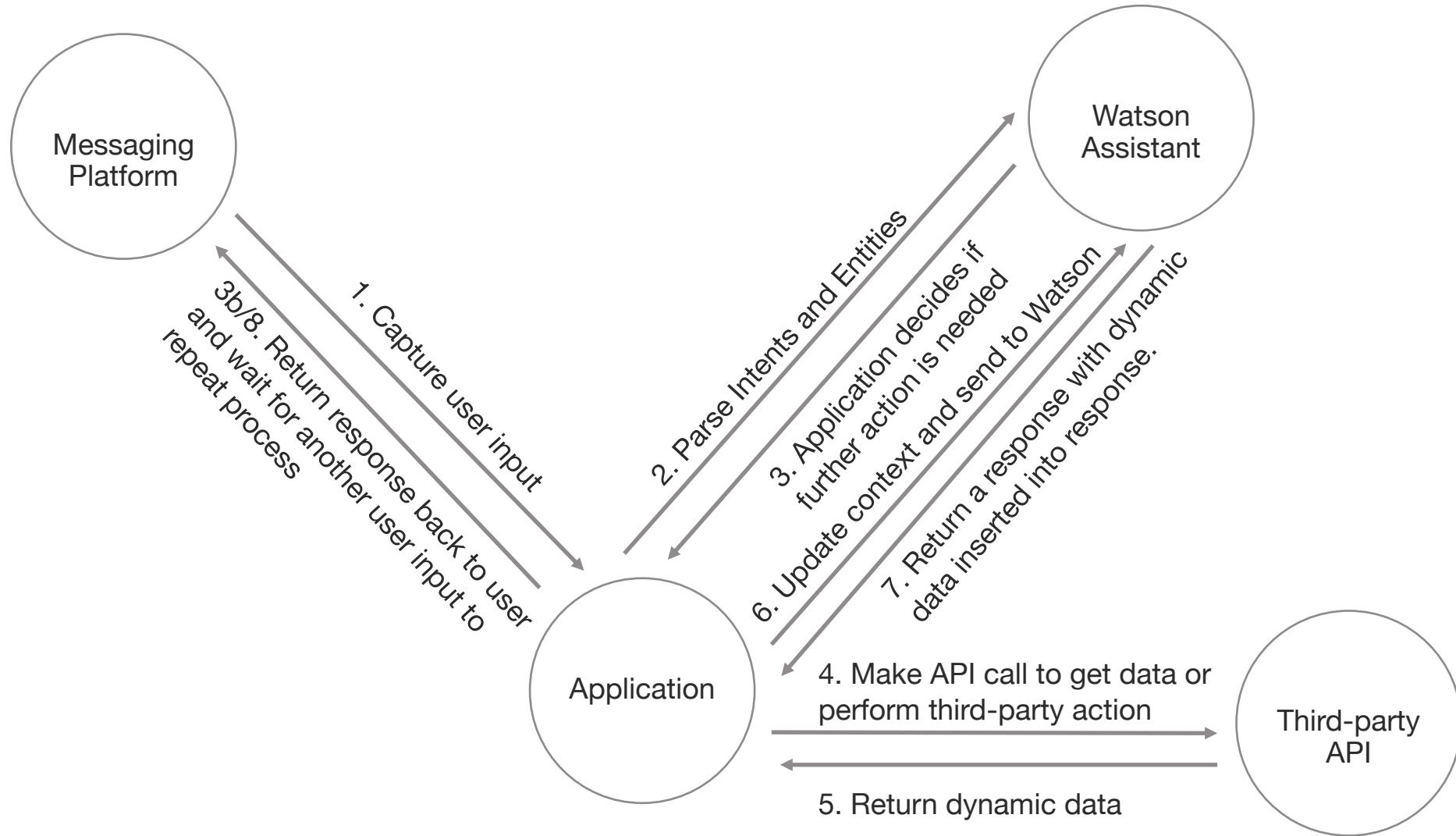


Online Store Chatbot



Simple Ranking Chatbot

Connecting Third-Party APIs via Client



Calling an IBM Cloud Function

Define actions that can make programmatic calls to external applications or services and get back a result as part of the processing that occurs within a dialog turn.

```
{  
  "context": {  
    "variable_name" : "variable_value"  
  },  
  "actions": [  
    {  
      "name": "<actionName>",  
      "type": "client | server",  
      "parameters": {  
        "<parameter_name>": "<parameter_value>",  
        "<parameter_name>": "<parameter_value>"  
      },  
      "result_variable": "<result_variable_name>",  
      "credentials": "<reference_to_credentials>"  
    }  
  ],  
  "output": {  
    "text": "response text"  
  }  
}
```