

BOTS, AI, AND JAVASCRIPT

Future of the Web with Conversational Interface



#IndexConf

Tomomi Imura (@girlie_mac)



tomomi imura

- advocate open web & technology
- code JavaScript & Node.js
- write & speak about tech
- hack useless stuff
- A cat lady of the InterWeb
- dev relations at Slack

**Talking to
non-human**



THINK

DENK

TAOIIIIS

मोचिए

PE



\$300,000

\$1,000,000

\$200,000

KEN


WATSON

BRAD

"Bots are like new applications that you can converse with. "

-- Satya Nadella, Microsoft



A photograph of Sundar Pichai, CEO of Google, standing on a stage during a presentation. He is wearing a dark jacket, glasses, and jeans, and is gesturing with his hands. Behind him is a large screen displaying the text "Mobile first to AI first". The stage features a black metal frame with green and blue geometric panels and several spotlights.

Mobile first to AI first

*“We will evolve
in computing
from a mobile
first to an AI first
world.”*

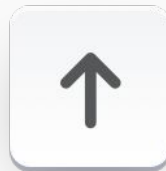
-- Sundar Pichai, Google




Traditional Web & App Interactions

hello | **Send**

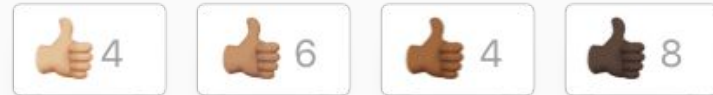
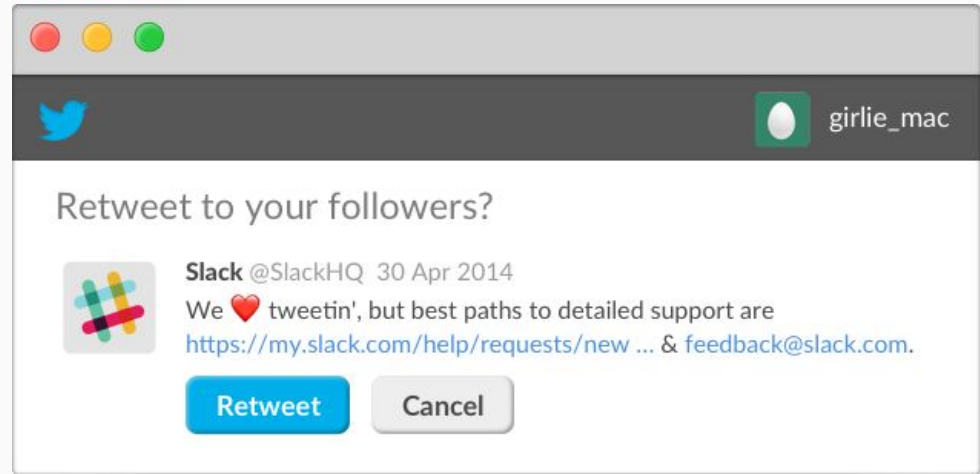
tab



Special Meals 

- Vegetarian lacto-ovo
- Hindu vegetarian**
- Vegan
- Gluten-free
- Kosher
- Muslim

Modern Web & Apps with Social Interactions







Conversational User Interactions: Siri and Alexa (Voice Assistants)



In various form-factors

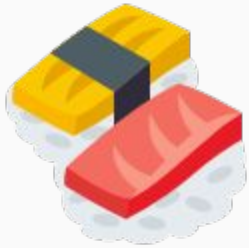
Conversational User Interactions for Kids - with Voice



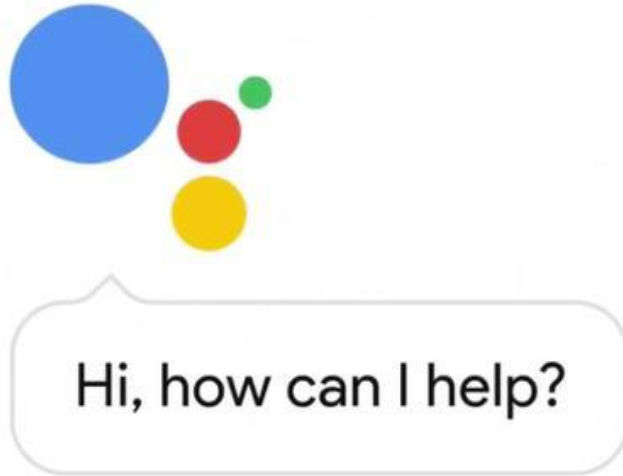
-  Tell Jokes
-  Answer Questions
-  Play Games
-  Make Stories

Conversational User Interactions in a robot shape - with Voice






Conversational User Interactions: Google Assistant (Voice & Text)

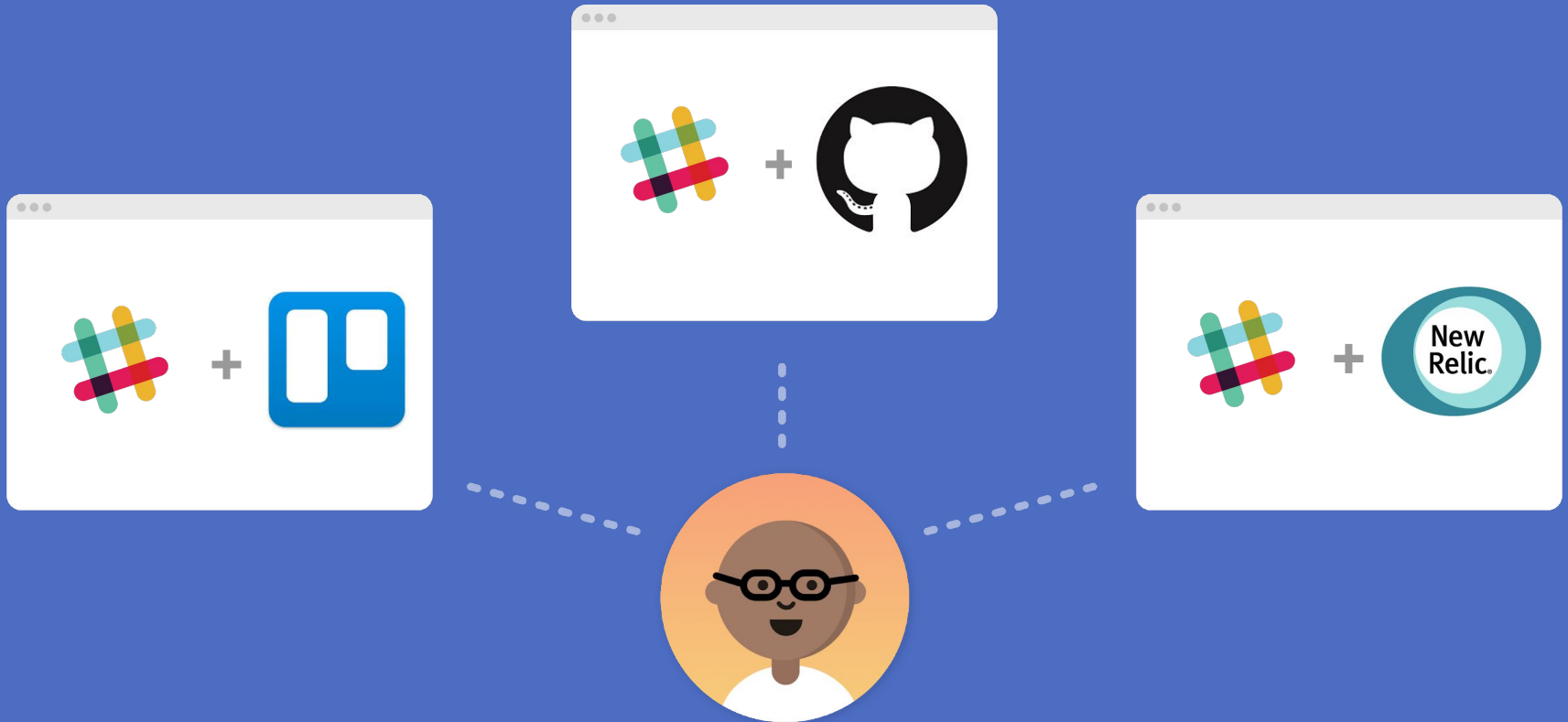


Conversational User Interactions: Slack Bots (Text)



 **Office Space Bot** 4:45 PM
Don't forget your TPS report!

Slack Integrations & Bots for Better Productivity



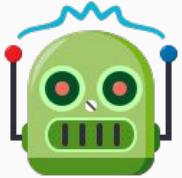
Graphic Interface to Conversational Interface

Deliver me a large margherita pizza!



What kind of crust do you want?

1. Regular crust
2. Thin crust
3. Gluten free crust



Thin crust

cha ching!



**Conversational
Interface achieves:**

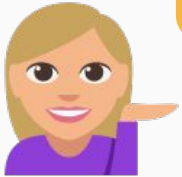
**Natural user
interactions with a
minimal visual
interface.**

**No UI Clutter.
Less Time Spent.**

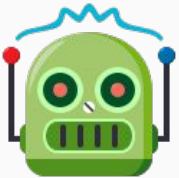
Come over to my place!

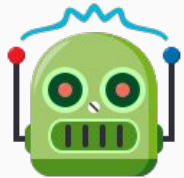
It's 325 Ruby Street.

Request a ride



Where is the address
of your home?





Your driver, Sam will pick you up in 6 minutes. Look for the red Toyota Prius!

Yes, get me a ride now



Alexa UX (“Voice Chrome” Examples)

Listening

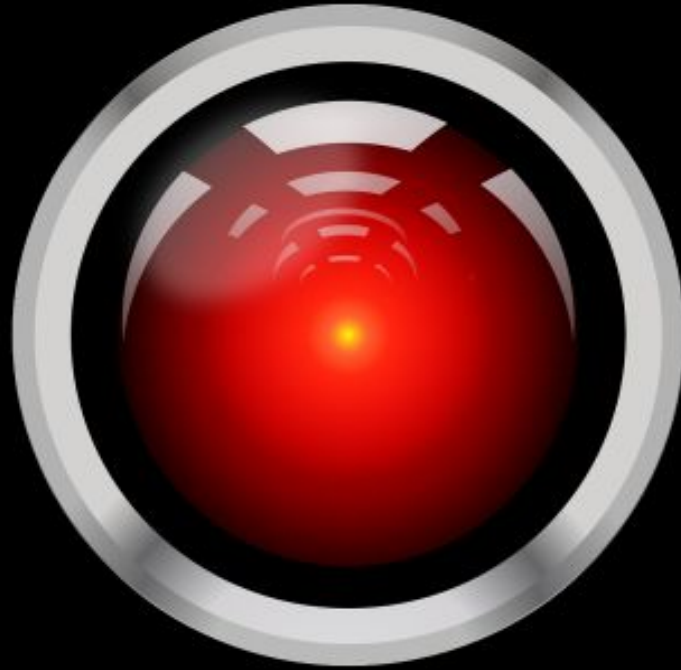


Speaking



Old concepts.

**Technology
caught up
with the
ideas.**



HAL9000: "I'm sorry Dave, I'm afraid I can't do that"



Siri: "I'm sorry Dave, I'm afraid I can't do that"

A futuristic hallway with a red carpet and a woman with a metallic headpiece. The woman is in the foreground, looking towards the right. She has a metallic, mesh-like headpiece that covers the top and back of her head, with a red, textured area visible on the side. Her neck and shoulders are also covered in a similar metallic mesh. In the background, another person is visible in the distance, and the hallway is lit with recessed lights. The overall atmosphere is clean and high-tech.

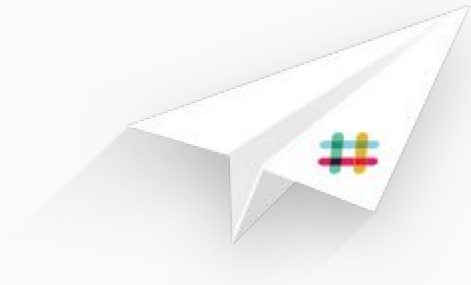
Nuh, we're still in control over AI. (I hope)

**Conversational
Interface is:**

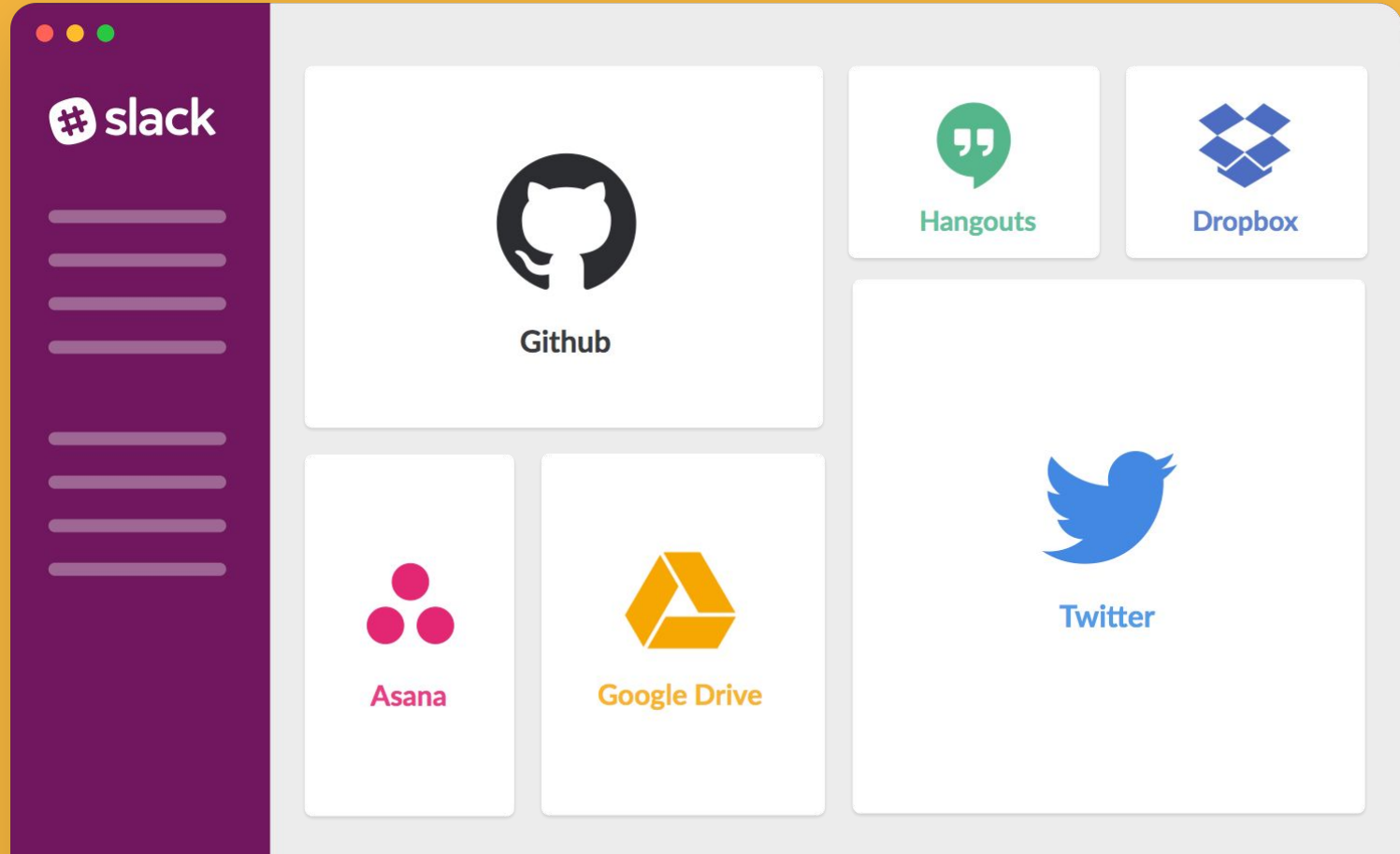
- **Intuitive**
- **Accessible**
- **Productive**

Messaging Platforms

- Slack
 - Facebook Messenger
 - Telegram
 - WeChat
 - Kik
 - Viber
 - LINE
- etc.



Messaging + Bots for More Interactive Communications





Stage Chicken APP 12:56 PM

Your code is being staged by dhruv in [stage 125545](#)

| [PR #66713](#) from apidocs_misc_bugs - timura

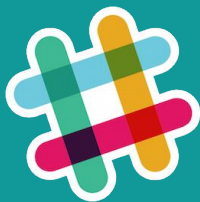


Deploy Chicken APP 1:00 PM

Your code is being deployed by kevin in [deploy 125546](#)

| [PR #66713](#) from apidocs_misc_bugs - timura

Slack bots at



JIRA Bot APP 2:05 PM

[PFA-489](#) updated by brenda:

Create documentation for new conversations.* endpoints

| **Epic Link** changed from [PFA-128](#) to [PFA-694](#)

Status changed from [In Development](#) to [Done](#)

Resolution changed to [Fixed](#)

👁 Only visible to you



Lunch Train APP 3:58 PM

You've started a lunch train! I've sent you controls via [@slackbot](#), conductor.



Lunch Train APP 3:58 PM

Chew choo! [@girlie_mac](#) started a train to Blue Bottle at 4:00 pm.
[@John Agan](#) is on board. Will you join?

Board the train

Slack bots at



Tomomi Imura JS 5:31 PM

(ノ ◦ ◦) ノ へ ー ー

Custom response



slackbot 5:31 PM

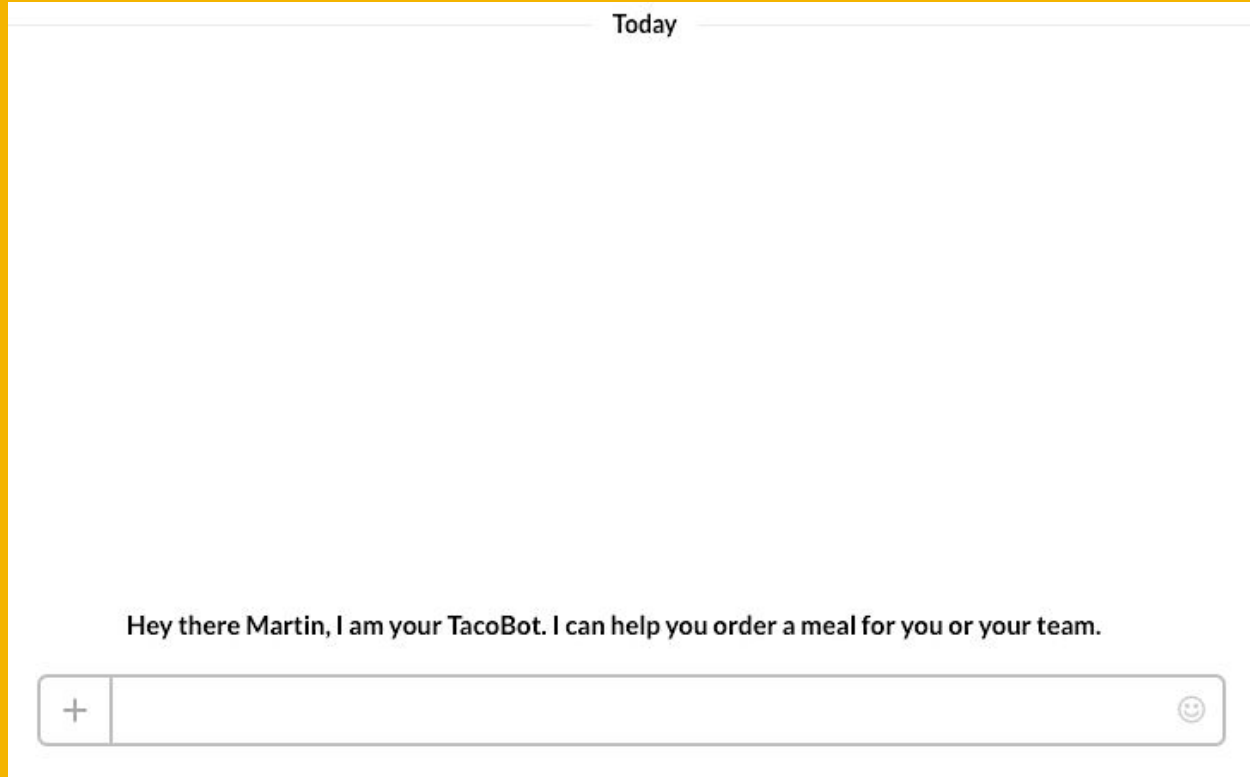
ー ー ノ (◦ _ ◦) - Please respect tables.



slackbot 3:00 PM



TacoBot by Taco Bell



<https://www.tacobell.com/feed/tacobot>

Natural Conversation APIs

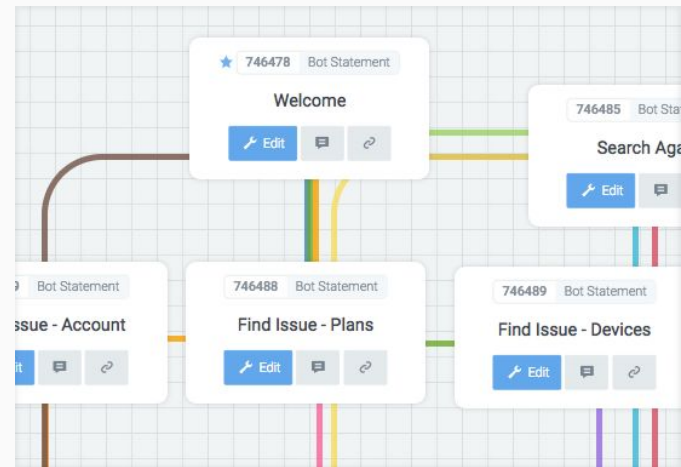
- DialogFlow (API.ai /Google)
- Wit.ai (Facebook)
- Microsoft Bot Framework
- Motion.ai
- Chatbots.io
- Converse AI
- Recast.ai
- Flow XO
- ManyChat
- Fluent.ai (Voice UI)

etc.

AlaaS

Artificial
Intelligence as
a Service 🤓

Motion.ai



One-click integrations

 Actions on Google <input type="checkbox"/>	 Web Demo <input type="checkbox"/>	 Facebook Messenger <input checked="" type="checkbox"/> SETTINGS	 Slack <input checked="" type="checkbox"/> SETTINGS
 Viber <input type="checkbox"/>	 Twitter <input type="checkbox"/>	 Twilio IP <input type="checkbox"/>	 Twilio (Text messaging) <input type="checkbox"/>
 Skype	 Tropo (Text messaging)	 Telegram	 Kik

DialogFlow (API.ai)

Hi Linda! Are you flying from San Francisco, as usual?

Hi,
I want to book a flight!



Yes, from SFO.

Where are you flying to?

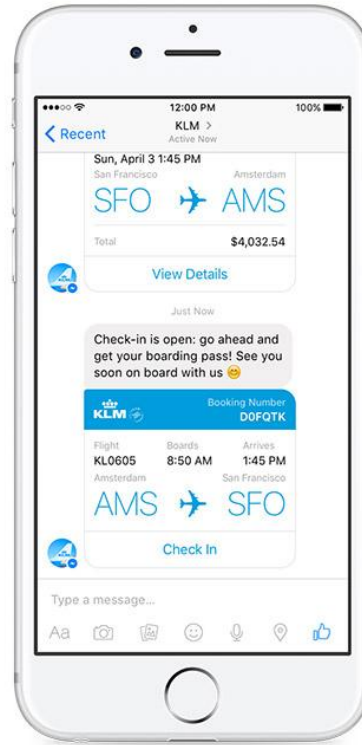
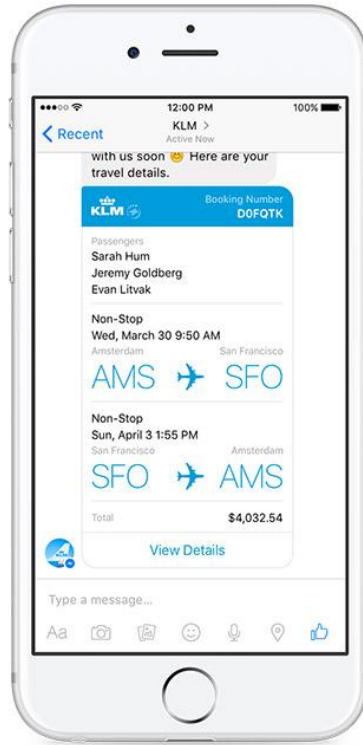
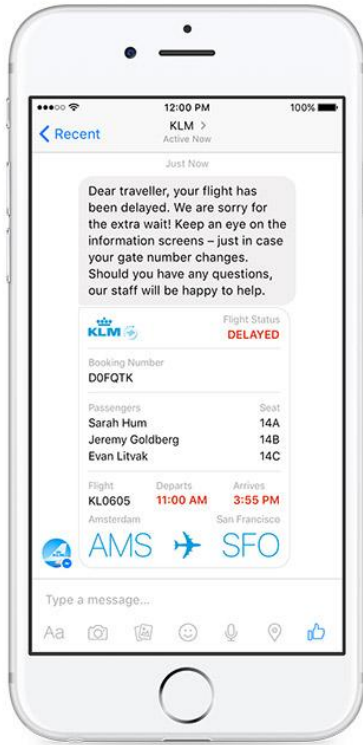
London



An airline customer service bot

Linda may not know she is talking to a bot!





NLP Platforms / APIs

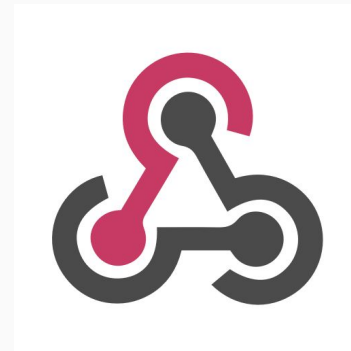
Natural Language Processing & Cognitive platforms:

- IBM Watson
- Google Cloud Natural Language API
- Microsoft LUIS
- Amazon Lex
- Baidu UNIT

Build Your Own Conversational Interface

**...with
JavaScript**

The APIs are
mostly
accessible with
JS



Example: API.ai + FB Messenger



```
const app = require('apiai')(CLIENT_ACCESS_TOKEN);
```

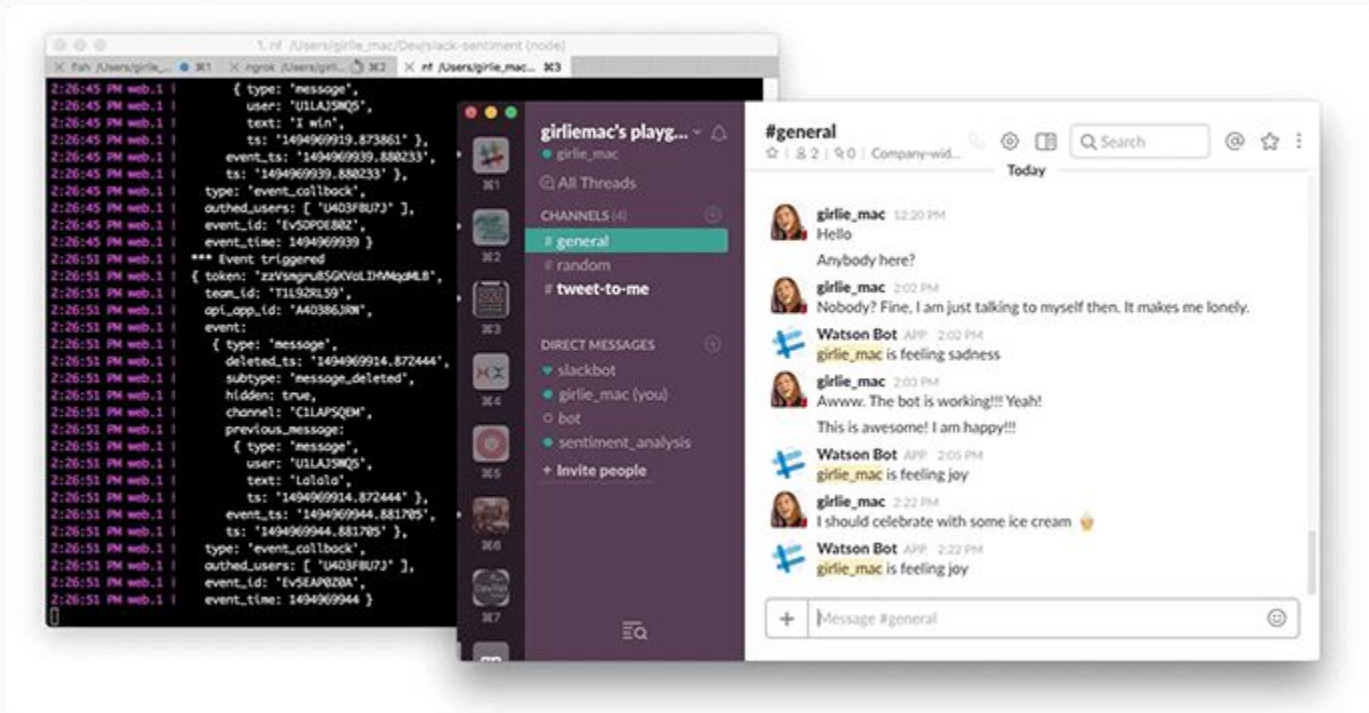
```
function sendMessage(event) {  
  let sender = event.sender.id;  
  let text = event.message.text;
```

API.ai Node.js SDK

```
  let ai = app.textRequest(text, {  
    sessionId: SESSION_STRING });
```

```
  ai.on('response', (response) => {  
    // Got a response. Let's POST to Facebook Messenger  
  });  
  ai.end();  
}
```

Example: IBM Watson + Slack



The image shows a terminal window on the left and a Slack chat window on the right. The terminal displays a sequence of JSON messages and event triggers. The Slack chat shows a conversation where a user named 'girlie_mac' says 'Hello' and 'Anybody here?'. A bot named 'Watson Bot' responds with 'Nobody? Fine, I am just talking to myself then. It makes me lonely.' The user then says 'Awww. The bot is working!!! Yeah!' and 'This is awesome! I am happy!!!'. The bot responds with 'girlie_mac is feeling joy' twice.

```
2:26:45 PM web.1 | { type: 'message',
2:26:45 PM web.1 |   user: 'U1LAJ5M0S',
2:26:45 PM web.1 |   text: 'I win',
2:26:45 PM web.1 |   ts: '1494969919.873861' },
2:26:45 PM web.1 |   event_ts: '1494969939.880233' },
2:26:45 PM web.1 |   ts: '1494969939.880233' },
2:26:45 PM web.1 |   type: 'event_callback',
2:26:45 PM web.1 |   outhed_users: [ 'U403FBU7J' ],
2:26:45 PM web.1 |   event_id: 'Ev50PDE802',
2:26:45 PM web.1 |   event_time: 1494969939 }
2:26:51 PM web.1 | *** Event triggered
2:26:51 PM web.1 | { token: 'zzV5sgnU8SGVotJHMAqgML8',
2:26:51 PM web.1 |   team_id: 'T1L9ZRL59',
2:26:51 PM web.1 |   api_app_id: 'A40386J0W',
2:26:51 PM web.1 |   event:
2:26:51 PM web.1 |     { type: 'message',
2:26:51 PM web.1 |       deleted_ts: '1494969914.872444',
2:26:51 PM web.1 |       subtype: 'message_deleted',
2:26:51 PM web.1 |       hidden: true,
2:26:51 PM web.1 |       channel: 'C1LAPSQEM',
2:26:51 PM web.1 |       previous_message:
2:26:51 PM web.1 |         { type: 'message',
2:26:51 PM web.1 |           user: 'U1LAJ5M0S',
2:26:51 PM web.1 |           text: 'Lalala',
2:26:51 PM web.1 |           ts: '1494969914.872444' },
2:26:51 PM web.1 |         event_ts: '1494969944.881705' },
2:26:51 PM web.1 |         ts: '1494969944.881705' },
2:26:51 PM web.1 |       type: 'event_callback',
2:26:51 PM web.1 |       outhed_users: [ 'U403FBU7J' ],
2:26:51 PM web.1 |       event_id: 'Ev5EAPQZBA',
2:26:51 PM web.1 |       event_time: 1494969944 }
```

girlie_mac's playg...

- girlie_mac
- All Threads
- CHANNELS (4)
- # general
- # random
- # tweet-to-me
- DIRECT MESSAGES
- slackbot
- girlie_mac (you)
- bot
- sentiment_analysis
- + Invite people

#general

Today

girlie_mac 12:20 PM
Hello

girlie_mac 2:02 PM
Anybody here?

girlie_mac 2:02 PM
Nobody? Fine, I am just talking to myself then. It makes me lonely.

Watson Bot APP 2:02 PM
girlie_mac is feeling sadness

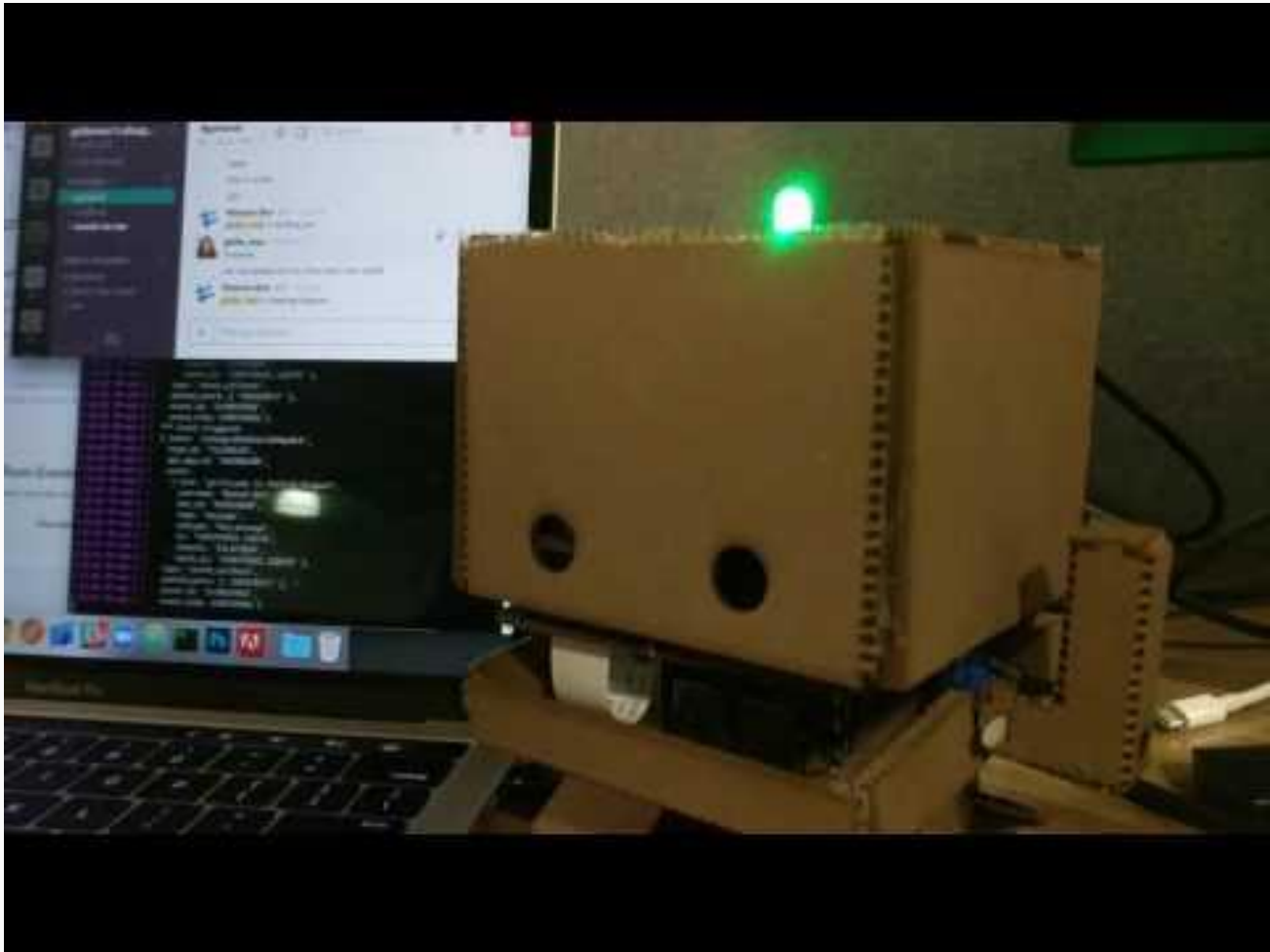
girlie_mac 2:03 PM
Awww. The bot is working!!! Yeah!

girlie_mac 2:03 PM
This is awesome! I am happy!!!

Watson Bot APP 2:05 PM
girlie_mac is feeling joy

girlie_mac 2:22 PM
I should celebrate with some ice cream 🍦

Watson Bot APP 2:22 PM
girlie_mac is feeling joy



The bot workflow:

1. Read each message on a Slack channel
2. Send the message to IBM Watson for examination
3. If the likelihood of an emotion is above the given confidence threshold post the most prominent emotion

Example: IBM Watson + Slack



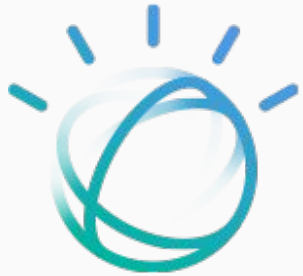
HTTP POST using ExpressJS

```
app.post('/events', (req, res) => {  
  let q = req.body;  
  
  if (q.type === 'event_callback') {  
    if(!q.event.text) return;  
    analyzeTone(q.event);  
  }  
});
```

Use Slack Events API to grab the text when a user post a message

Pass the text data to Watson to analyze

Example: IBM Watson + Slack



```
const watson = require('watson-developer-cloud');
let tone_analyzer = watson.tone_analyzer({
  username: process.env.WATSON_USERNAME,
  password: process.env.WATSON_PASSWORD,
});

const confidencethreshold = 0.55;
tone_analyzer.tone({text: text}, (err, tone) => {
  tone.document_tone.tone_categories.forEach((tonecategory) => {
    if(tonecategory.category_id === 'emotion_tone'){
      tonecategory.tones.forEach((emotion) => {
        if(emotion.score >= confidencethreshold) {
          postEmotionOnSlackChannel(emotion);
        }
      });
    }
  });
});
```

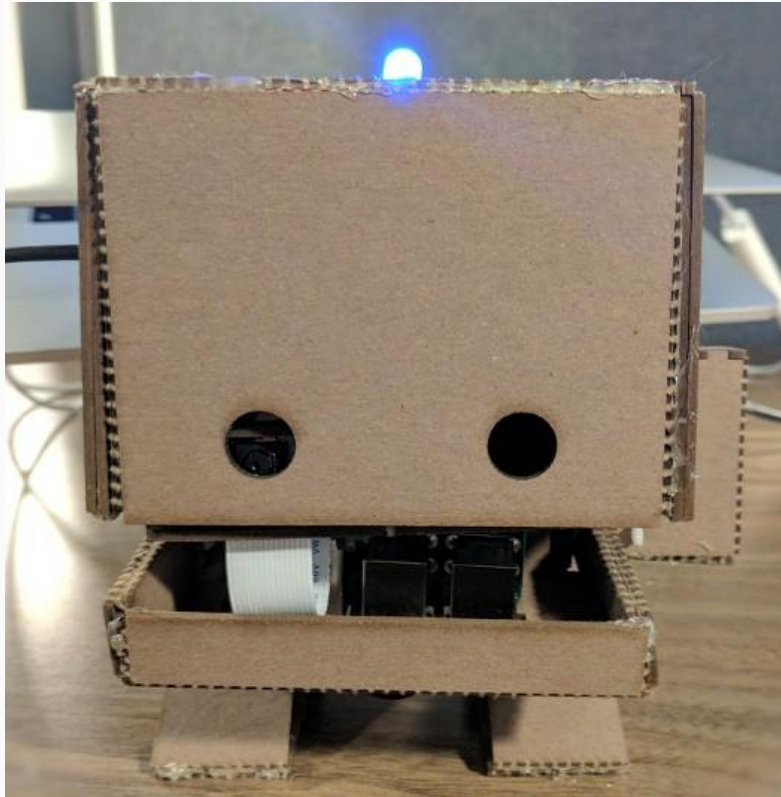
Returns emotions score in 0 to 1

Just initializing it w/
your API credentials

Post the result on Slack



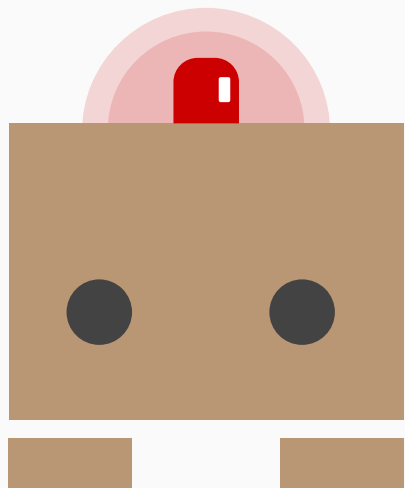
Example: IBM Watson + Slack + Raspberry Pi (for fun)



```
function colorEmotion(emotion) {  
  if (emotion.tone_id === 'anger') {  
    setLED(red);  
  } else if(emotion.tone_id === 'joy') {  
    setLED(yellow);  
  } else if(emotion.tone_id === 'fear') {  
    setLED(purple);  
  } else if(emotion.tone_id === 'disgust') {  
    setLED(green);  
  } else if(emotion.tone_id === 'sadness') {  
    setLED(blue);  
  }  
}
```



Change the LED color to match the emotion



An angry customer detected. Connect the customer with a human!

Ack, this sucks! I want my money back!

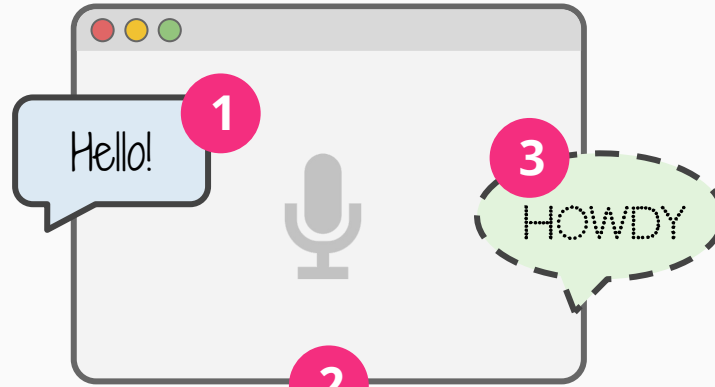


Conversational Interface with Voice in Browser? -----

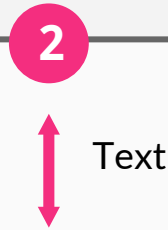
Project: Artificial Voice Chat

1. User talk to browser

Voice command:
Voice -> Text



3. Browser speaks back
Text -> Synthetic Voice



2. Generate Artificial reply
Using the 3rd party API

Web Speech API

1 Speech Recognition & 3 Speech Synthesis



<http://caniuse.com/#feat=speech-recognition>

<http://caniuse.com/#feat=speech-synthesis>

Web Speech API: Speech Recognition

```
const SpeechRecognition =  
window.SpeechRecognition || window.webkitSpeechRecognition;
```

In the current Chromium, it is still prefixed

```
const recognition = new SpeechRecognition();
```

Get an instance of the SpeechRecognition, the controller interface

Web Speech API: Speech Recognition (Cont'd)

```
recognition.lang = 'en-US';  
recognition.interimResults = false;
```

Some properties

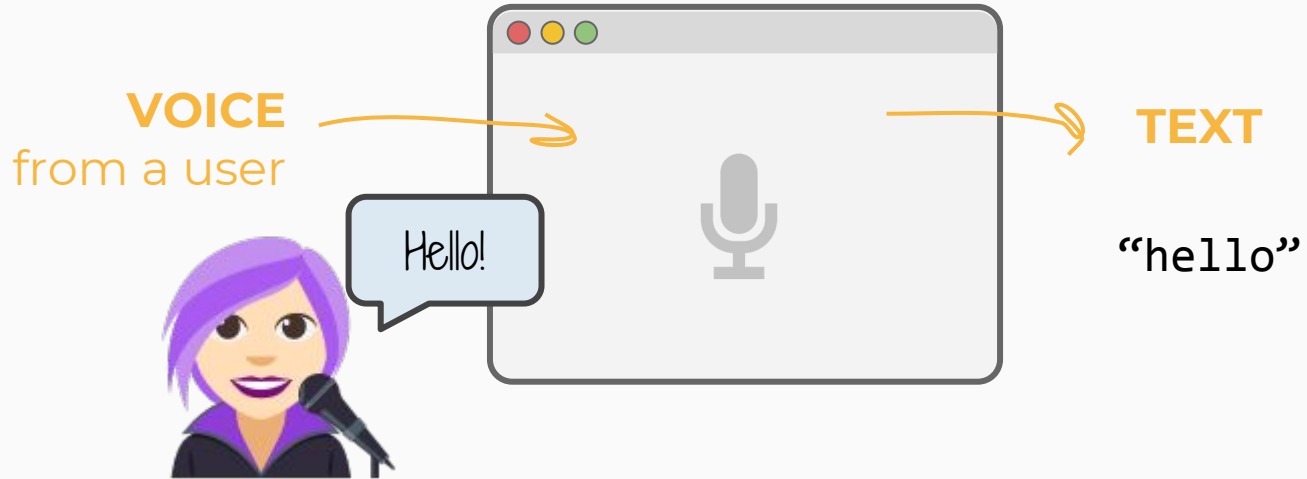
```
recognition.start();
```

Methods: start(), stop(), abort()

```
recognition.addEventListener('result', (e) => {  
  let last = e.results.length - 1;  
  let text = e.results[last][0].transcript;  
});
```

Events: onresult, onerror, onaudiostarted, onaudioend, etc.

Web Speech API: Speech Recognition



Web Speech API: Speech Synthesis

```
const synth = window.speechSynthesis;
```

← No vendor prefix 😊

```
const utterance = new SpeechSynthesisUtterance();
```

```
utterance.text = 'I am a robot';
```

```
utterance.pitch = 1.5;
```

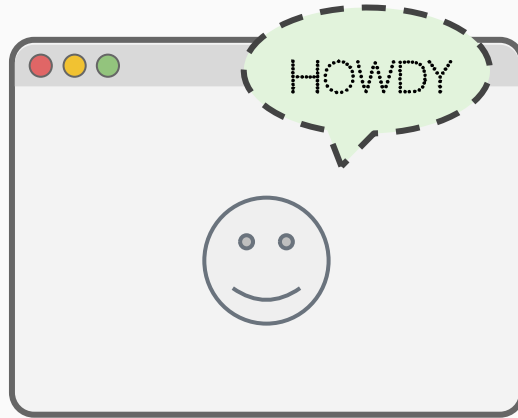
```
utterance.voice = 'Alex';
```

← Properties of the SpeechSynthesisUtterance interface

```
synth.speak(utterance);
```

← Get available voices with `synth.getVoices()` method

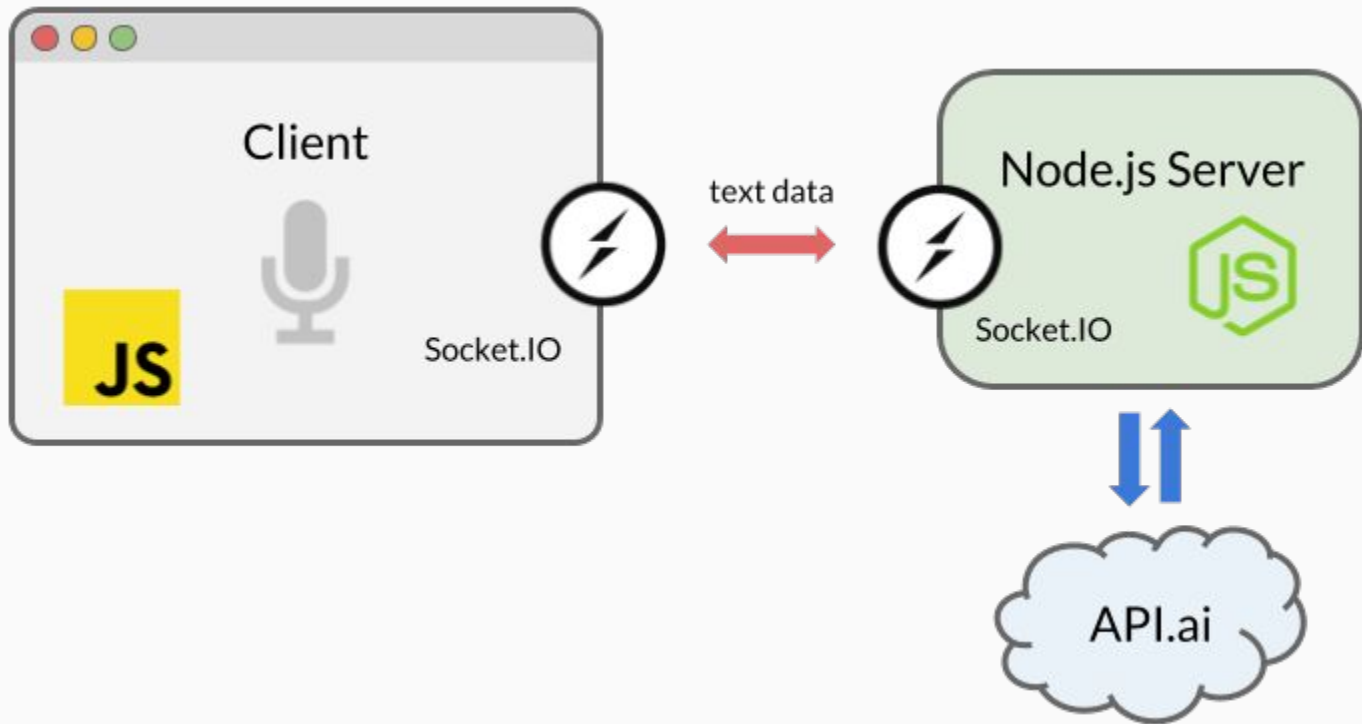
Web Speech API: Speech Synthesis



TEXT to VOICE

“howdy”

(voice by Alex or
whoever browser is)



Demo on Chrome

<https://webspeech.herokuapp.com/>

Building A Simple AI Chatbot With Web Speech API And Node.js

By Tomomi Imura

August 7th, 2017 Accessibility, API, UI

Using voice commands has become pretty ubiquitous nowadays, as **more mobile phone users use voice assistants** such as Siri and Cortana, and as devices such as Amazon Echo and [Google Home](#) have been invading our living rooms. These systems are built with speech recognition software that allows their users to issue [voice commands](#). Now, our web browsers will become familiar with to Web Speech API, which allows users to integrate voice data in web apps.

Search on Smashing Magazine

e.g. JavaScript Search

Smashing Newsletter

Subscribe to our email newsletter for useful tips and valuable resources, sent out every second Tuesday.

email address Subscribe

234,898 Subscribers powered by MailChimp

Design Systems

As designing static pages has become untenable, many have started to approach design in a modular way. In this book, we'll identify what makes an **effective design system** that empowers teams to create great digital products. [Pre-order the book now](#) →

<https://www.smashingmagazine.com/2017/08/ai-chatbot-web-speech-api-node-is/>

**Conversational
Interface is
for human.**

**A bots interface really is
a human interface.**

It looks like you're writing a letter.

Would you like help?

- Get help with writing the letter
- Just type the letter without help
- Don't show me this tip again



Remember me?

Clippy 1997 - 2007

Thank you!

[@girlie_mac](#)

girliemac.com

github.com/girliemac

speakerdeck.com/girlie_mac





Attribution:

Open Emoji by [Emoji-One](#) (CC-BY 4.0)