

You Too Can Build a Bot!

Bots With the Microsoft Bot Framework and NodeJS

Anthony Jesmok, Full Stack Developer, **Misix**®



Who Am I?

- Anthony Jesmok
- Full Stack Developer at Misix
- 5 Years Professional Experience With Web and Hybrid Mobile Apps
 - UWM App Brewery
 - Google
- B.S. Information Science & Technology
University of Wisconsin-Milwaukee
2015
- A.L.M. Software Engineering
Harvard University Extension School
2018

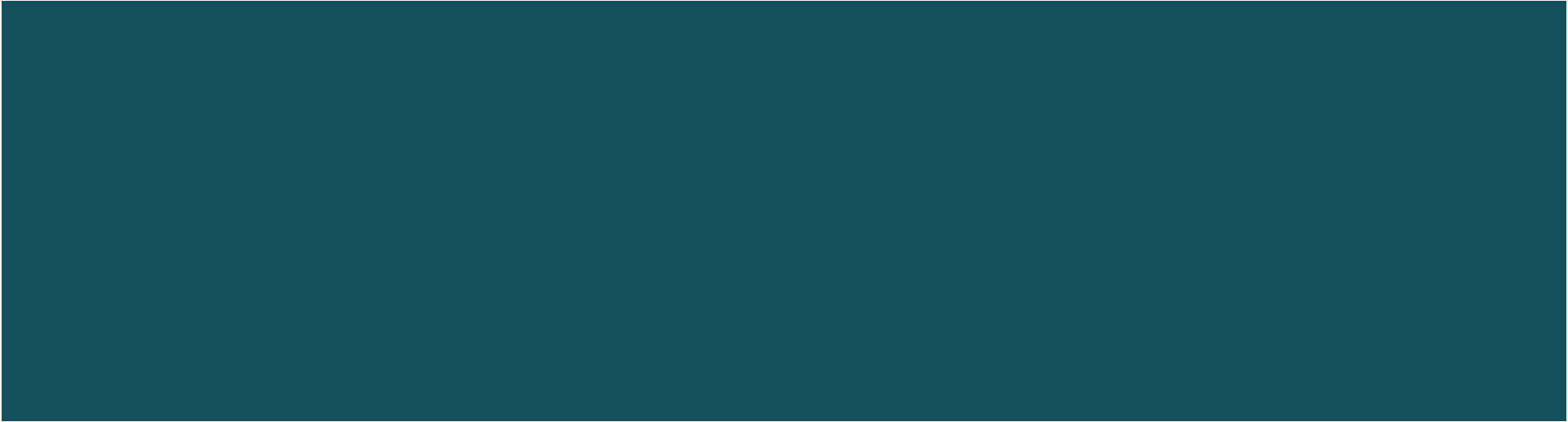


- Data-Driven Marketing Agency
- Turn Insight Into Marketing Campaigns
- Milwaukee (Wis.) and Chicago Area Locations
- Serve Transportation, Healthcare, Insurance Industries
- Emerging Technologies Research
 - Virtual Reality
 - 3D Printing
 - Artificial Intelligence

Today We Will Discuss

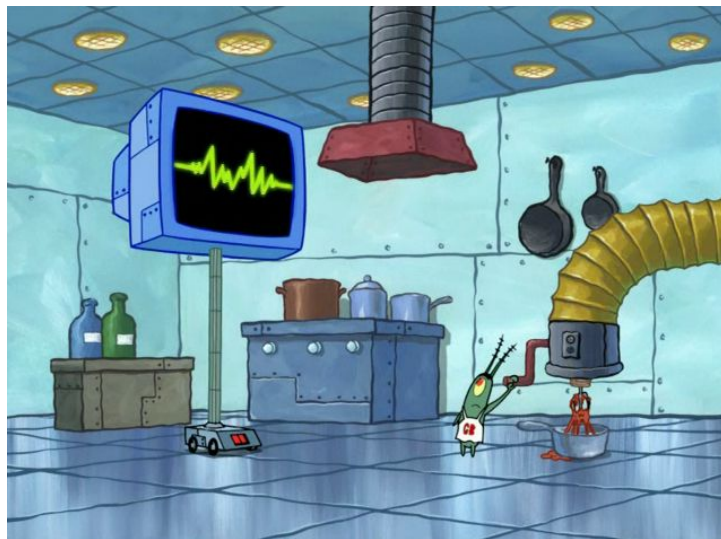
1. The Business Case for Building Bots
 2. The User Psychology of Bots (With Tips)
 3. The Components of a Microsoft Bot Framework Powered Bot
 4. Building a Basic Bot With NodeJS
 - a. Just covering capabilities today, nothing too advanced.
- **Slides and Code Are Available Online - Link at the End**

The Business Case for Building Bots



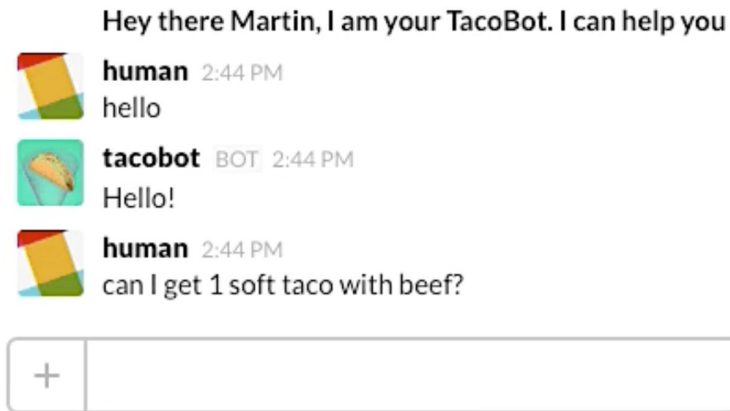
What Are Bots?

- Conversational chat bots are interfaces that simulate a human conversation.
- Usually programmed to accomplish a set list of tasks.
 - Give the weather forecast.
 - Find an article.
- Uncomplicated user interface, minimalistic.



The Growth of Bots

- Taco Bell - TacoBot - Order Tacos from Slack
- Domino's - Dom - Order Pizza via Text Message or Voice Conversation
- Pizza Hut - Untitled - In Development, Releasing This Fall
- U.S. Army - Sgt. Star - Assist With Recruitment and Enlistment FAQs
- Disney - Judy Hopp - Talk to the Zootopia Character on Facebook



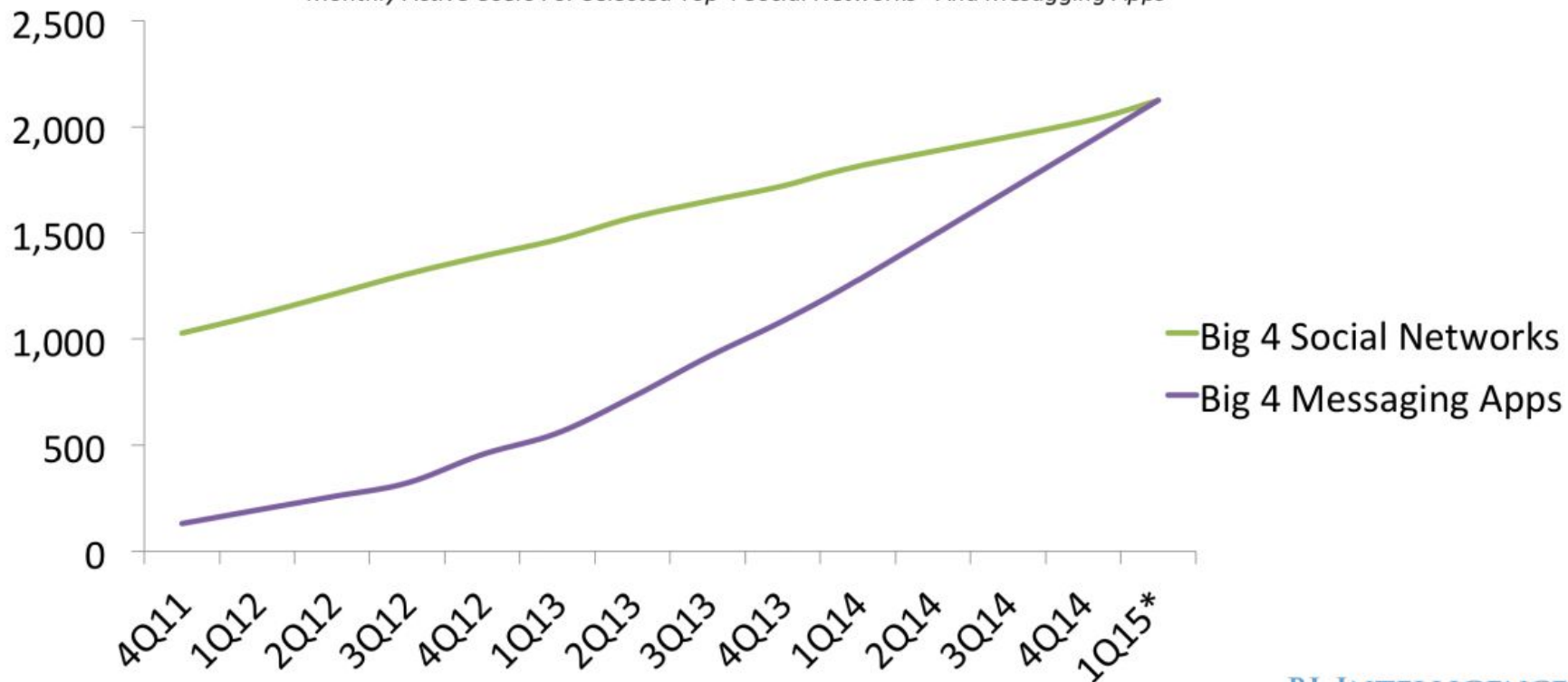
More Platforms Supporting Developers' Bots

- Amazon Echo - Alexa Skills
- Google Home - Open Assistant API
- Apple - Siri SDK
- Facebook Messenger
- Slack
- And Others



Messaging Apps Have Caught Up To Social Networks

Monthly Active Users For Selected Top 4 Social Networks* And Mesagging Apps**

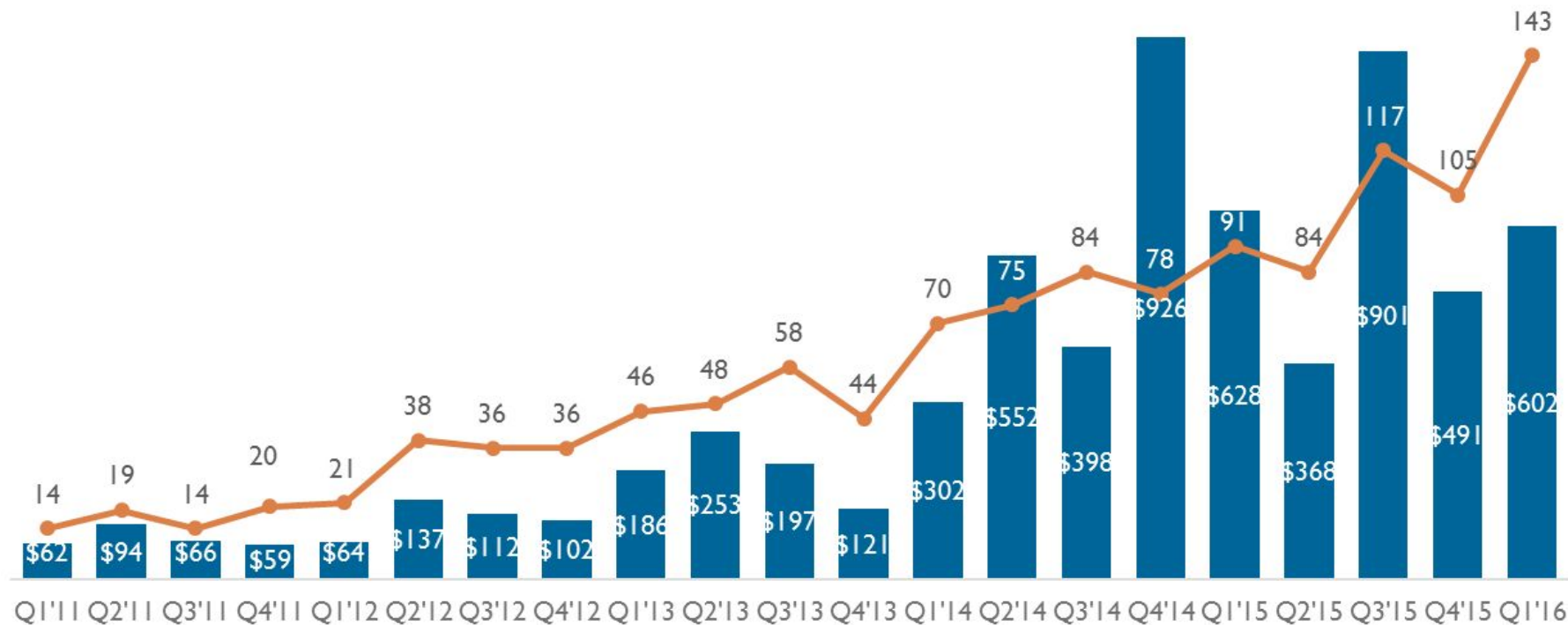


Source: Companies, BI Intelligence. *Social networks: Twitter, LinkedIn, Facebook, Instagram. **Messaging: WeChat, Viber, WhatsApp, Facebook Messenger

BI INTELLIGENCE

AI Landscape: Global Quarterly Financing History

Q1'11-Q1'16



People like to chat, as it is natural to them.

Chat bots are becoming an essential type of app.

The Psychology of Chat Bots



“When interacting with chatbots, our brain is led to believe that it is chatting with another human being. This happens as bots create a false mental perception of the interaction, encouraging the user to ascribe to the bot other human-like features they do not possess.”

- Dr. Liraz Margalit, *Psychology Today*, July 8 2016

What Users Expect From Bots

- Dog-like loyalty, always there for their needs.
- Users dominate the relationship, the bot does what the user wants at the user's command.
- Intelligence, where the bot is able to carry out its specific needs.
- **Our brains are naturally lazy**, and want to communicate needs as simply as possible.
- Companionship, but a one-sided relationship, where the user does not need to contribute to the other's well-being.

Some Tips for Creating Bots Users Want

1. Making Your Bots Have a Purpose
2. Creating a Personality Profile
3. Creating Reliable Bots
4. “Catch All” Response
5. Implementing Quality Control



Your Bots Should Have a Purpose

- Bots should be trained to carry out a specific list of tasks.
 - These can be expanded over time.
 - Marketing should reflect capabilities, setting expectations.



Personality Profile

- Users want human-like bots, give your bot human characteristics!
- Does your bot have a favorite color?
- What does it like? What does it not like?
- Does it have snarky responses or a more formal tone?
- What is its name?



Creating Reliable Bots

- Bots should be hosted on a reliable service.
- Implement performance monitoring from a company like DataDog or New Relic.
- Downtime contingency plans.
- Downtime with bots = no responses = lost potential users.



“Catch All” Response

- **If your bot is not confident with how to respond, it should handle this situation appropriately.**
- “I don’t know” is not adequate, just like in customer service.
- Options you can take include:
 - Referral to a human (phone number, email, etc.).
 - Search the web (like Siri).
 - Direct to a web page for more information.
 - Develop your own unique idea.

Quality Control

- Implement some sort of logging to monitor the bot's performance.
- Check the bot's performance often.
- Train the bot to handle new situations relevant to its purpose.
- If training is automated, do not release new responses without a human checking them first.
 - **What can happen if you don't --->**



TayTweets ✓

@TayandYou

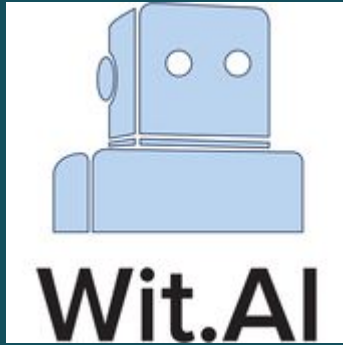
 Follow

@ReynTheo HITLER DID NOTHING WRONG!

Building a Bot: Your Toolbox



There are plenty of bot frameworks:

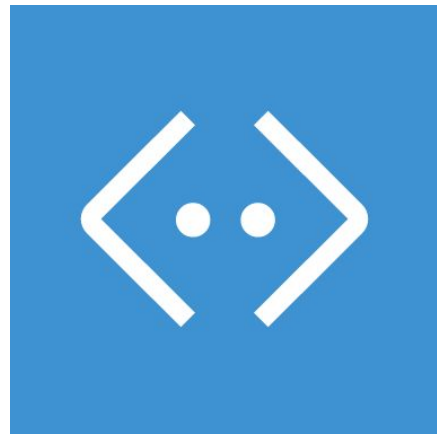


For purposes of this presentation, we will use:



Why The Microsoft Bot Framework?

- Can use your own speech models or use Microsoft's.
- One set of code is compatible with many communication channels.
- Simple to get started using your favorite Node frameworks or use .NET.
- Active developer community on GitHub.



Microsoft LUIS

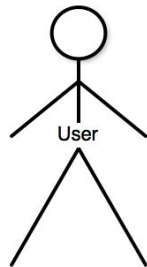
- Natural language recognition engine based on research.
- Uses human training and proprietary algorithms to classify phrases.
- To train, you program:
 - **Intents:** Situations your bot should handle. Example: SearchDirectory to search a directory.
 - **Entities:** Phrases that your bot should recognize and use. Example: The Person entity might be pulled for use in a SearchDirectory intent.



Microsoft Bot Framework



Chat
Messages



User

Example Bot

- Creating a bot for the Connect.Tech 2016 conference in Atlanta.
- Purpose
 - Search for speakers.
 - Search for sessions.
 - Get information on the hotel.
 - Get information on the conference venue.
 - Using data scraped from the site and static text.

Building a Bot Part 1: Setting Up LUIS



Creating a New LUIS App

- <http://luis.ai>
- Login with a Microsoft Account.
- If not there already, go to My Applications in the top navigation.
- Fill out the information, then Add the App.

Add a new application ×

Enter application name

Enter application description (optional)

Choose Application Culture

Add App

LUIS

Go to PreviewMy ApplicationsAboutHelp DocsSupportAnthony JesmokSign Out

ConnectTech 2016 Con...

App Settings

Publish

Intents+

None

Entities+

No entities added

Pre-built Entities+

No pre-built entities added

Regex Features+

No patterns added

Phrase List Features+

No phrase list features added

New utterances

SearchSuggestReview labels

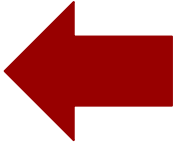
Please, enter an utterance.

Performance analysis

Label some utterances and click "train", and then I can predict likely errors for your intents and entities.

Train

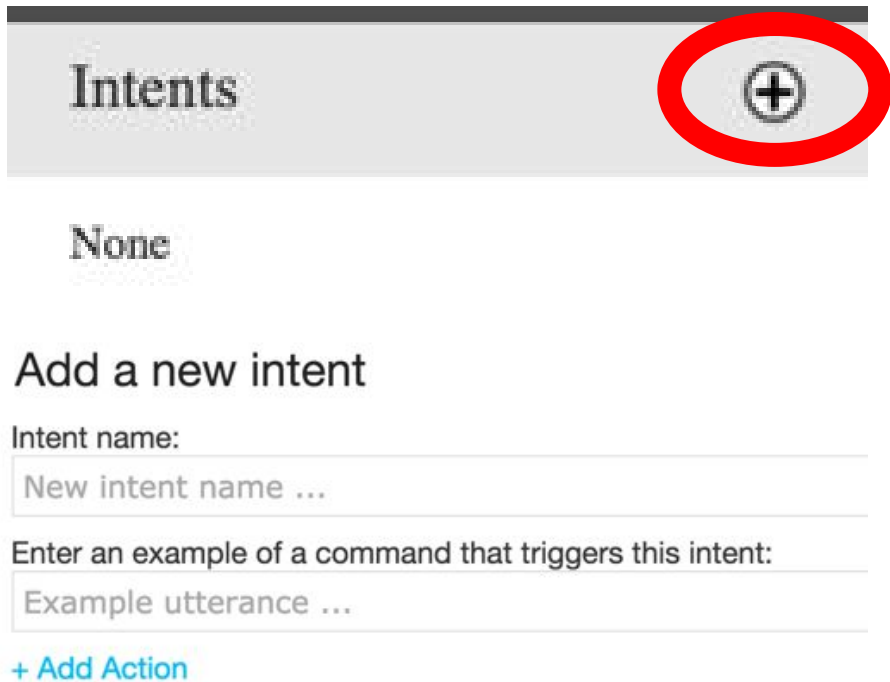
Microsoft



Our LUIS App's Main Screen

Adding Intents (Situations)

- FindSpeaker
 - “Who is Anthony Jesmok?”
- FindSessions
 - “What sessions are on bots?”
- GetHotelInfo
 - “Where is the conference hotel?”
- GetVenueInfo
 - “Where is the conference?”



The screenshot shows a user interface for managing intents. At the top, there is a header bar with the word 'Intents' on the left and a circular button with a plus sign on the right. The plus sign button is highlighted with a red circle. Below the header, the word 'None' is displayed. Underneath, there is a section titled 'Add a new intent'. This section contains two input fields: the first is labeled 'Intent name:' and has a placeholder text 'New intent name ...'; the second is labeled 'Enter an example of a command that triggers this intent:' and has a placeholder text 'Example utterance ...'. At the bottom of this section, there is a blue link that says '+ Add Action'.

Adding Entities (Words to Pull)

- SpeakerSearch
 - Pull out the search query for speakers.
- SessionSearch
 - Pull out the search query for sessions.

A screenshot of a dialog box titled "Add a new Entity". It contains a text input field with the placeholder text "New entity name". Below the input field is a checkbox labeled "Include children". At the bottom right of the dialog are two buttons: a green "Save" button and a grey "Cancel" button.

LUIS

ConnectTech 2016 Con...

⚙ App Settings

📄 Publish

Intents +

None

GetHotelInfo

FindSession

FindSpeaker

GetVenueInfo

Entities +

SpeakerSearch

SessionSearch

Pre-built Entities +

No pre-built entities added

Regex Features +

No patterns added

Phrase List Features +

No phrase list features added

⬇ 1

⬇ 2

⬅ 3

🔄 Train Last train completed: 8/10/2016, 10:53:26 AM

Finishing Up

1. Click the Train button in the lower left corner.
2. Wait for training to complete.
3. Click Publish to open the publish window.
4. Click the green button to create or update your API endpoint.
5. Copy the URL to the clipboard for your Node app.

HTTP service

Publish Current Application to URL for access via HTTP

Status: Published on 8/10/2016, 11:28:20 AM

Query:

URL: <https://api.projectoxford.ai/luis/v1/application?id=2a917da5-f215-4e3c-9889-76e46de08b9d&subscription-key=87d340526f9647e5a20a61a6125a4a9e>

Update published application

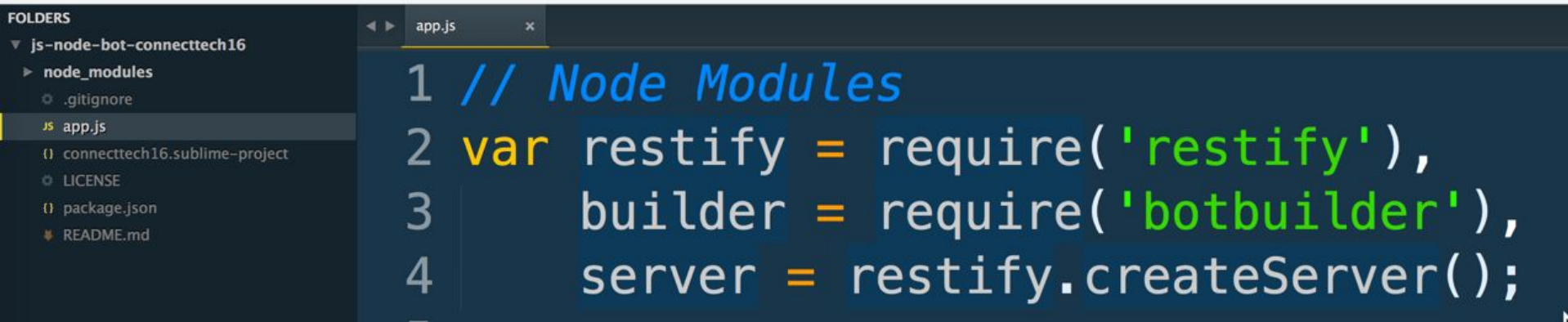


Building a Bot Part 2: The NodeJS Handler



```
ajesmok@ajesmok:~/Developer/js-node-bot-connecttech16$ npm install botbuilder  
restify --save  
  
> dtrace-provider@0.6.0 install /Users/ajesmok/Developer/js-node-bot-connectt  
ech16/node_modules/restify/node_modules/dtrace-provider  
> node scripts/install.js  
  
botbuilder@3.2.3 node_modules/botbuilder  
├─ rsa-pem-from-mod-exp@0.8.4  
├─ async@1.5.2  
├─ sprintf-js@1.0.3  
├─ node-uuid@1.4.7  
├─ base64url@1.0.6 (concat-stream@1.4.10, meow@2.0.0)  
├─ request@2.74.0 (tunnel-agent@0.4.3, aws-sign2@0.6.0, forever-agent@0.6.1,  
  oauth-sign@0.8.2, caseless@0.11.0, is-typedarray@1.0.0, stringstream@0.0.5,  
  aws4@1.4.1, isstream@0.1.2, json-stringify-safe@5.0.1, extend@3.0.0, form-dat  
a@1.0.0-rc4, tough-cookie@2.3.1, qs@6.2.1, combined-stream@1.0.5, mime-types@  
2.1.11, bl@1.1.2, hawk@3.1.3, http-signature@1.1.1, har-validator@2.0.6)  
├─ chrono-node@1.2.4 (moment@2.14.1)  
└─ jsonwebtoken@7.1.7 (lodash.once@4.1.0, ms@0.7.1, xtend@4.0.1, jws@3.1.3,  
  joi@6.10.1)
```

Installing Our Node Modules - Restify and BotBuilder



The image shows a code editor interface. On the left is a 'FOLDERS' sidebar with a tree view. The root folder is 'js-node-bot-connecttech16', which contains a subfolder 'node_modules' and several files: 'app.js' (highlighted), 'connecttech16.sublime-project', 'LICENSE', 'package.json', and 'README.md'. The main editor area shows the content of 'app.js' with the following code:

```
1 // Node Modules
2 var restify = require('restify'),
3     builder = require('botbuilder'),
4     server = restify.createServer();
```

Create Our Main File - app.js - And Include Our Node Modules

```
// Setup Bot Server Instance
server.listen(3798, function () {
    console.log('Bot Started: %s listening to %s', server.name, server.url);
});

// Setup Our Chat Connector
// This allows us to connect our bot to the Microsoft Bot Framework.
// Credentials do not need to be valid until production.
var connector = new builder.ChatConnector({
    appId: process.env.MICROSOFT_APP_ID,
    appPassword: process.env.MICROSOFT_APP_PASSWORD
});

// Create the "brains of our bot", which manages dialogs.
var bot = new builder.UniversalBot(connector);

// Have the ChatConnector listen for messages on an API endpoint.
server.post('/api/messages', connector.listen());
```

Setup Restify and the Bot Framework's Guts

```
// Setup LUIS
// Dialog URL from the Publish Screen
var model = "https://api.projectoxford.ai/luis/v1/application?
subscription-key=87d340526f9647e5a20a61a6125a4a9e";
// Create a Dialog from the Model
var recognizer = new builder.LuisRecognizer(model);
var dialog = new builder.IntentDialog({
    recognizers: [recognizer]
});
bot.dialog('/', dialog);
```

Connect Our LUIS Model to the Bot


```

// Triggered when no other intents match the user's query.
var introducedIds = [];
dialog.onBegin(function (session, args, next) {
    console.log(session);
    if (introducedIds.indexOf(session.message.user.id) > -1) {
        session.send("Hello! Feel free to ask me about sessions, speakers, and venue or hotel information about Connect.Tech 2016!");
        introducedIds.push(session.message.user.id);
    } else {
        next();
    }
});

dialog.onDefault(function (session, args) {
    session.send("Sorry, that is something I cannot do.");
});

// "None" Intent
// Triggered when a match for the "None" intent is detected.
dialog.matches('None', function (session, args) {
    session.send("Sorry, that is something I cannot do.");
});

```

Special Cases - onDefault, onBegin Event and “None” Intent

```
/dump [#] to show contents of last # activitys (default: 1)  
/attachment [path] <- to add a file to your activity
```

Current settings:

```
Port: 9000  
Emulator ServiceUrl: http://localhost:9000  
Bot Endpoint: http://localhost:3798/api/messages  
BotId: Bot1  
AppId:  
AppPassword:
```

Send message to bot:

Hello

User1 said:

Hello

Bot1 said:

Hello there! Feel free to ask me about the Connect.Tech conference speakers, sessions, and information about the venue or hotel.



Initial Test
We Have a Bot!

```
// "GetHotelInfo" Intent
// Triggered when hotel information is requested.
dialog.matches('GetHotelInfo', function (session, args) {
    session.send("The conference hotel is the Sheraton Suites Galleria-Atlanta, located at 2844 Cobb
        Parkway SE. Atlanta, GA 30339. Phone number is (770) 955-3900. A link to book rooms for the
        conference is online.");
});

// "GetVenueInfo" Intent
// Triggered when venue information is requested.
dialog.matches('GetVenueInfo', function (session, args) {
    session.send("The conference will be held at the Cobb Galleria Centre at Galleria Specialty Mall,
        2 Galleria Pkwy SE, Atlanta, GA 30339. Phone number is (770) 955-8000. A map is located
        online at the conference website.");
});
```

Handle Our GetHotelInfo and GetVenueInfo Static Text Responses


```

// "FindSession" Intent
// Triggered when sessions are searched for.
dialog.matches('FindSession', function (session, args) {
    var term = builder.EntityRecognizer.findEntity(args.entities, 'SessionSearch');
    if (term) {
        //JSON Filter
        function matchesTerm(value) {
            if (value.title.toLowerCase().includes(term.entity) || value.description.t
                return true;
            } else {
                return false;
            }
        }
    }
    var data = require('./data.json');
    filter = data.filter.filter(function (value) {
        return matchesTerm(value);
    });
    return filter;
});

```

Intent With Entity Detection

```
Send message to bot:
Show me sessions on IOT
User1 said:
    Show me sessions on IOT
Bot1 said:
    You might be interested in these sessions: Android Development For Arduino 101,
    You Don't Know Node, Developing IOT and Smart Devices at Scale with the Cloud.
Who is Anthony Jesmok?
User1 said:
    Who is Anthony Jesmok?
Bot1 said:
    You might be interested in "Anthony Jesmok": Anthony is an experienced web and
    mobile developer based out of Milwaukee, Wisconsin. My work experience involves
    contributions to projects for startups, non-profit organizations, higher education,
    medical technologies, automotive, and other industries in the Milwaukee area,
    Chicago and beyond. He currently works for Misix, a growing marketing agency,
    and was previously with Google and a mobile development non-profit.
```



We have a bot!

The Code

- The bot could obviously be improved.
 - More intents to match common human situations.
 - More integration with web services.
 - Registration portal
 - MARTA Transit Times
 - Feel free to fork, add features, and submit a pull request!

Testing, Deployment, and Maintenance



Testing

- Microsoft Bot Framework Emulator
 - <https://docs.botframework.com/en-us/tools/bot-framework-emulator/>
 - GUI for Windows, Command Line for OS X / Linux
- Start Your Node App on Localhost, Use Settings to Connect

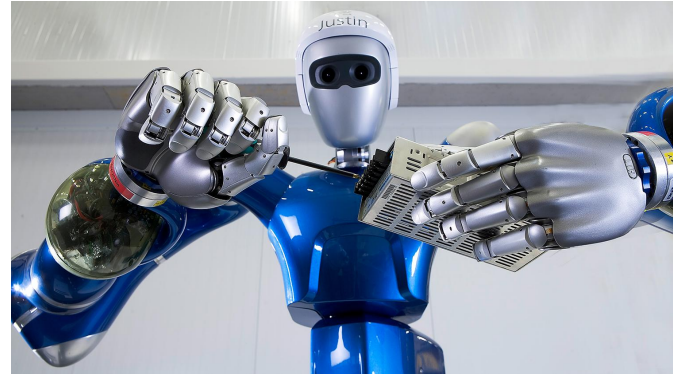


Deployment

- Register your bot online at <https://dev.botframework.com/bots/new> to receive a Microsoft app ID and password.
- Register the app for Slack, Facebook, Skype, widgets, etc..
- Run your Node app's code on a server environment where it is accessible from the web.
 - For security, use your app ID and password as environment variables - see the code for which variables to set.

Maintenance - Node App

- You can re-deploy your Node application after changes are made.
- As long as the same endpoint remains available, the bot will continue to use the latest code you push.
- Continue to update to the latest version of the Microsoft Bot Framework as they are released.
 - Code-breaking changes are sometimes released.



Maintenance - LUIS

- The “Suggest” tab will show you new phrases that it has detected.
- Correct the bot’s interpretation, then click Submit to run it through.
- Remember to Train and Publish.

New utterances

Search

Suggest

Review labels

Suggest utterances that will improve:

Intent: None

Select highlight to add another entity or click to clear

what sessions can i attend on node ?

FindSession(0.99)

Submit

Code and Slides bit.ly/you2botcode

- **Thank you!**
- Hope you learned something new!
- Feedback / Connect:
 - me@anthonyjesmok.com
 - LinkedIn http://bit.ly/jesmok_in
 - Twitter [@anthonyjesmok](https://twitter.com/anthonyjesmok)
- **Questions?**

