An Introduction to a Lot of Things, Including Botkit

Sam Havens

April 2016

Plan

I. Who I am

II. Conversational UI

III. Botkit

IV. Dependencies & Setup

V. Contact & Thanks

VI. Q&A

VII. Build

INTRODUCTION

Hi, I’m Sam

This is me

- I used to teach math and science at various levels
- I studied physics and math
- For the last two years I have been a developer at CarLabs
    - We make tools to help normal people research cars effectively
    - Backend: PHP on Yii2/MySQL with some Node.js
    - Frontend: jQuery/ES5 transitioning to React/ES6
- I have a really cute daughter, Poppy

CONVERSATIONAL UI

What and Why

Context: Other Computing Interfaces

- **CLI:** The command line interface (CLI) is the most direct way to interact with a computer. You type commands, which, if valid, are executed.

Context: Other Computing Interfaces

- **CLI:** The command line interface (CLI) is the most direct way to interact with a computer. You type commands, which, if valid, are executed.
  - Pros: powerful, fast, scriptable

Context: Other Computing Interfaces

- **CLI:** The command line interface (CLI) is the most direct way to interact with a computer. You type commands, which, if valid, are executed.
  - Pros: powerful, fast, scriptable
  - Cons: inflexible, unforgiving, hard to learn

Context: Other Computing Interfaces

- **CLI:** The command line interface (CLI) is the most direct way to interact with a computer. You type commands, which, if valid, are executed.
    - Pros: powerful, fast, scriptable
    - Cons: inflexible, unforgiving, hard to learn
- **GUI:** Graphical user interface (GUI) is the most common metaphor for interacting with computers. Invented at Stanford/Xerox in the 70s. WIMP: windows, icons, menus, & pointing device.

Context: Other Computing Interfaces

- **CLI:** The command line interface (CLI) is the most direct way to interact with a computer. You type commands, which, if valid, are executed.
  - Pros: powerful, fast, scriptable
  - Cons: inflexible, unforgiving, hard to learn
- **GUI:** Graphical user interface (GUI) is the most common metaphor for interacting with computers. Invented at Stanford/Xerox in the 70s. WIMP: windows, icons, menus, & pointing device.
  - Pros: easy to learn, forgiving

Context: Other Computing Interfaces

- **CLI:** The command line interface (CLI) is the most direct way to interact with a computer. You type commands, which, if valid, are executed.
    - Pros: powerful, fast, scriptable
    - Cons: inflexible, unforgiving, hard to learn
- **GUI:** Graphical user interface (GUI) is the most common metaphor for interacting with computers. Invented at Stanford/Xerox in the 70s. WIMP: windows, icons, menus, & pointing device.
    - Pros: easy to learn, forgiving
    - Cons: you have to look at a screen (texting while driving? the blind?), does not do well with hypotheticals, conditionals and vagueness (what is the WIMP metaphor for “If I have more than enough in my checking for bills and expenses, automatically transfer whatever I can afford to savings each day”?)

Conversational UI

- **CUI:** A conversational user interface (CUI) allows a user to interact with a computer using human language, in either speech or text. A CLI can be regarded as a CUI with a very poor conversationalist on the other end. Examples: Siri, Cortana, Google Now, Alexa, x.ai, Quartz (app)

Conversational UI

- **CUI:** A conversational user interface (CUI) allows a user to interact with a computer using human language, in either speech or text. A CLI can be regarded as a CUI with a very poor conversationalist on the other end. Examples: Siri, Cortana, Google Now, Alexa, x.ai, Quartz (app)
    - Pros: No learning curve, no application-specific commands, low friction (“Please download our app!” vs “sms this number”), accessibility, hypotheticals, conditionals, and vagueness

Conversational UI

- **CUI:** A conversational user interface (CUI) allows a user to interact with a computer using human language, in either speech or text. A CLI can be regarded as a CUI with a very poor conversationalist on the other end. Examples: Siri, Cortana, Google Now, Alexa, x.ai, Quartz (app)
  - Pros: No learning curve, no application-specific commands, low friction (“Please download our app!” vs “sms this number”), accessibility, hypotheticals, conditionals, and vagueness
  - Cons: New (less resources, best practices not yet established), understanding natural language is hard, forces things into a linear narrative (search has to be perfect because exploration is miserable - imagine being read a list of your recent debit card transactions vs scanning that list)

Conversational UI

Note: I will use the terms “conversational UI” and “bot” pretty much interchangeably.

Conversational UI

Note: Benedict Evans, an *amazing* technology writer, wrote about Conversation UI a few days ago and I just read it - great stuff.

Why?

Reducing friction is magic: People could write about themselves and connect with each other on the internet for at least a decade before Facebook and Twitter, but nothing exciting happened until it became very easy to do. CUI lowers the difficulty of tasks by removing all steps beyond “say what you want to happen.”

Why?

Reducing friction is magic: People could write about themselves and connect with each other on the internet for at least a decade before Facebook and Twitter, but nothing exciting happened until it became very easy to do. CUI lowers the difficulty of tasks by removing all steps beyond “say what you want to happen.”

Smaller screens: Phones, watches, etc. We want more information in less space... language is incredibly information dense per cm^2. Especially voice.

Why?

Reducing friction is magic: People could write about themselves and connect with each other on the internet for at least a decade before Facebook and Twitter, but nothing exciting happened until it became very easy to do. CUI lowers the difficulty of tasks by removing all steps beyond “say what you want to happen.”

Smaller screens: Phones, watches, etc. We want more information in less space... language is incredibly information dense per cm^2. Especially voice.

IoT: The “Internet of Things” is coming - “smart” thermostats, fridges, lightbulbs, dog collars, cars... are you going to have a different app for everything you own, and learn its settings? Or will Alexa/Siri/whatever interface with all that for you, and you just tell her what you want?

Why Now?

NLP: There have been large advances in Natural Language Processing (NLP) in the last few years using machine learning techniques, so it is finally plausible to try to understand what someone intends from what they say.

Why Now?

NLP: There have been large advances in Natural Language Processing (NLP) in the last few years using machine learning techniques, so it is finally plausible to try to understand what someone intends from what they say.

Power: People carry around tiny, powerful computers with them everywhere.

Why Now?

NLP: There have been large advances in Natural Language Processing (NLP) in the last few years using machine learning techniques, so it is finally plausible to try to understand what someone intends from what they say.

Power: People carry around tiny, powerful computers with them everywhere.

Useful: Phones, watches, Iot. A CUI is more useful on a watch than a 27” monitor.

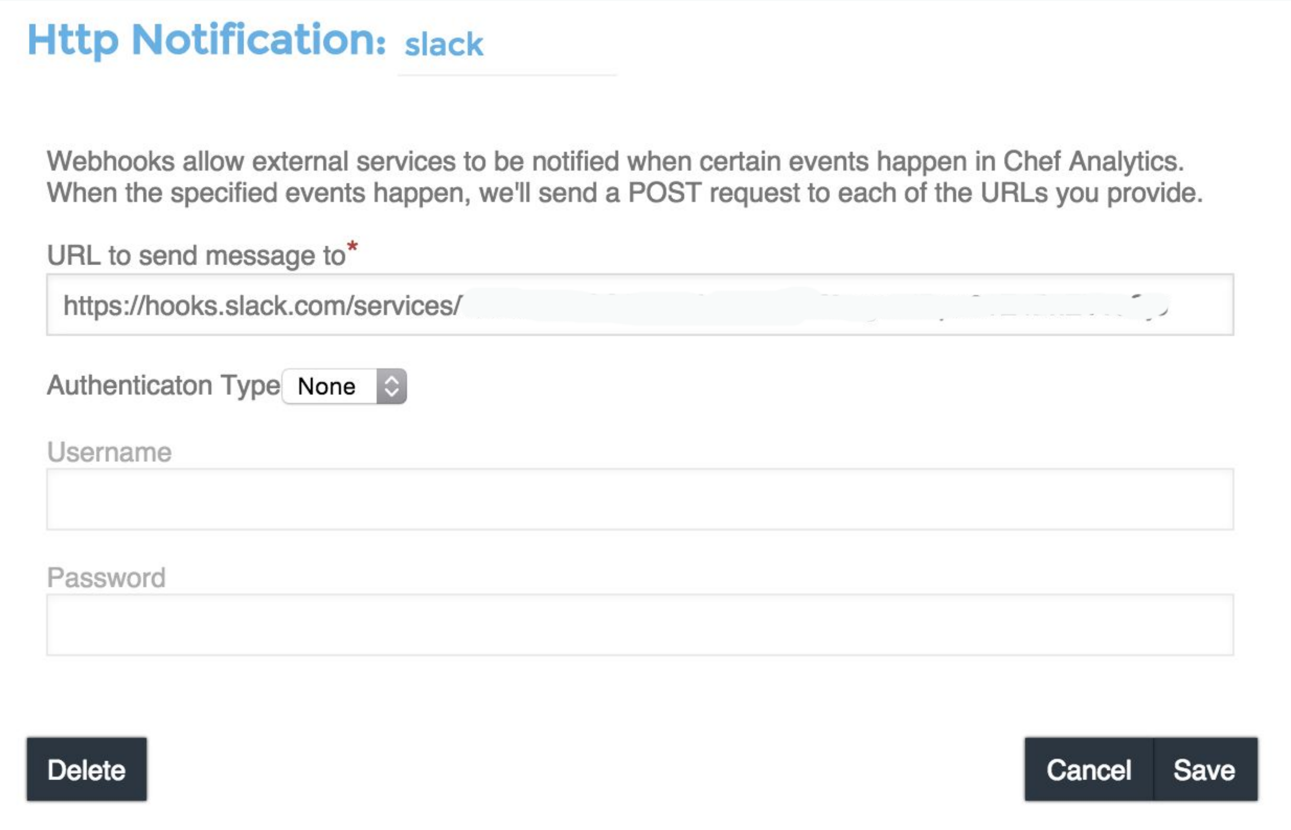
www.fraps.com

BOTKIT

A Framework for Slack Bots

Slack is awesome

Integrations make Slack even more awesome



Http Notification: slack

Webhooks allow external services to be notified when certain events happen in Chef Analytics. When the specified events happen, we'll send a POST request to each of the URLs you provide.

URL to send message to*

https://hooks.slack.com/services/[redacted]

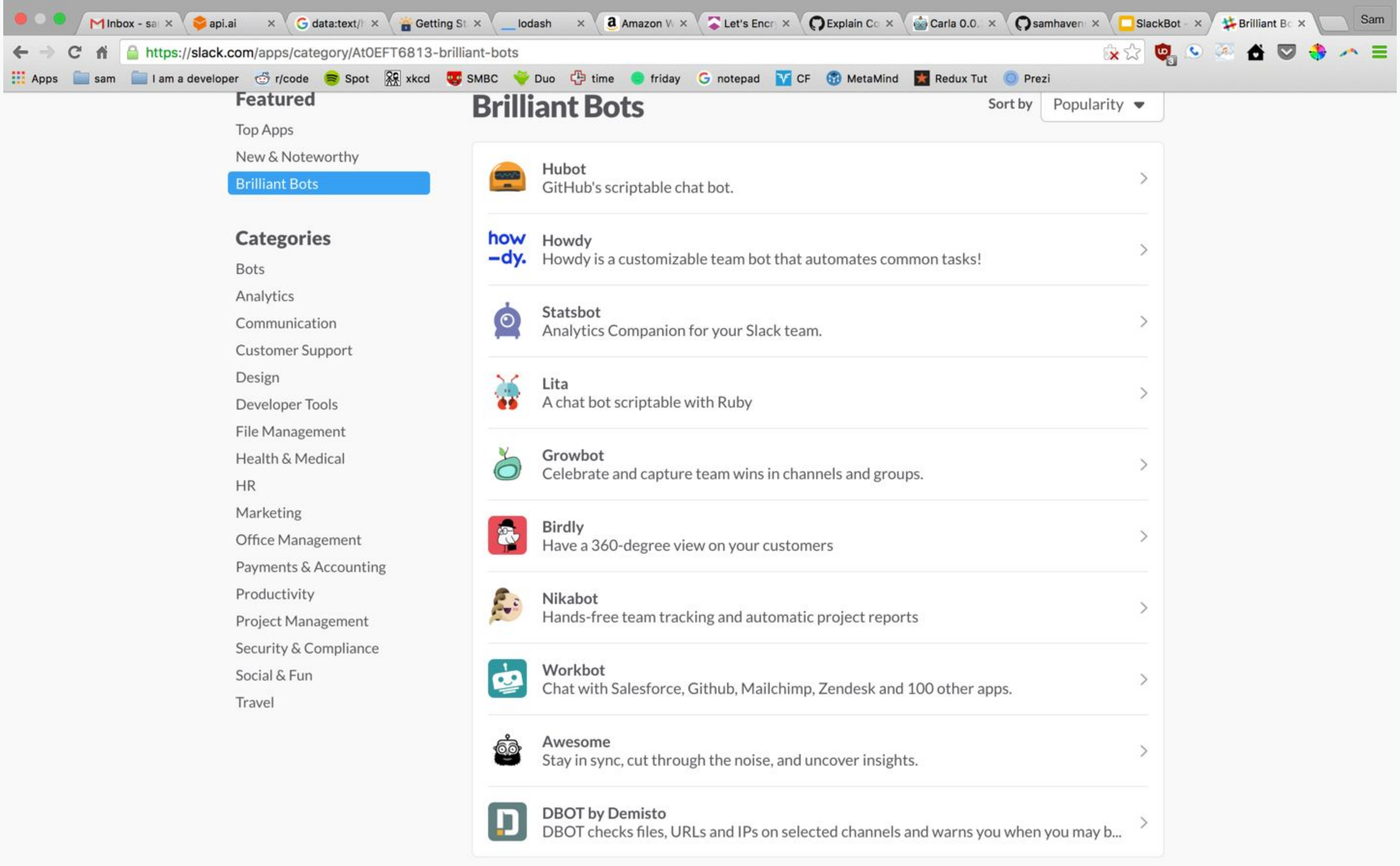
Authenticaton Type None

Username

Password

Bot integrations are fantastic

I was going to say bots are the most awesome part of Slack, but honestly... giphy integration ftw



Brilliant Bots

Hubot
GitHub's scriptable chat bot.

Howdy
Howdy is a customizable team bot that automates common tasks!

Statsbot
Analytics Companion for your Slack team.

Lita
A chat bot scriptable with Ruby

Growbot
Celebrate and capture team wins in channels and groups.

Birdly
Have a 360-degree view on your customers

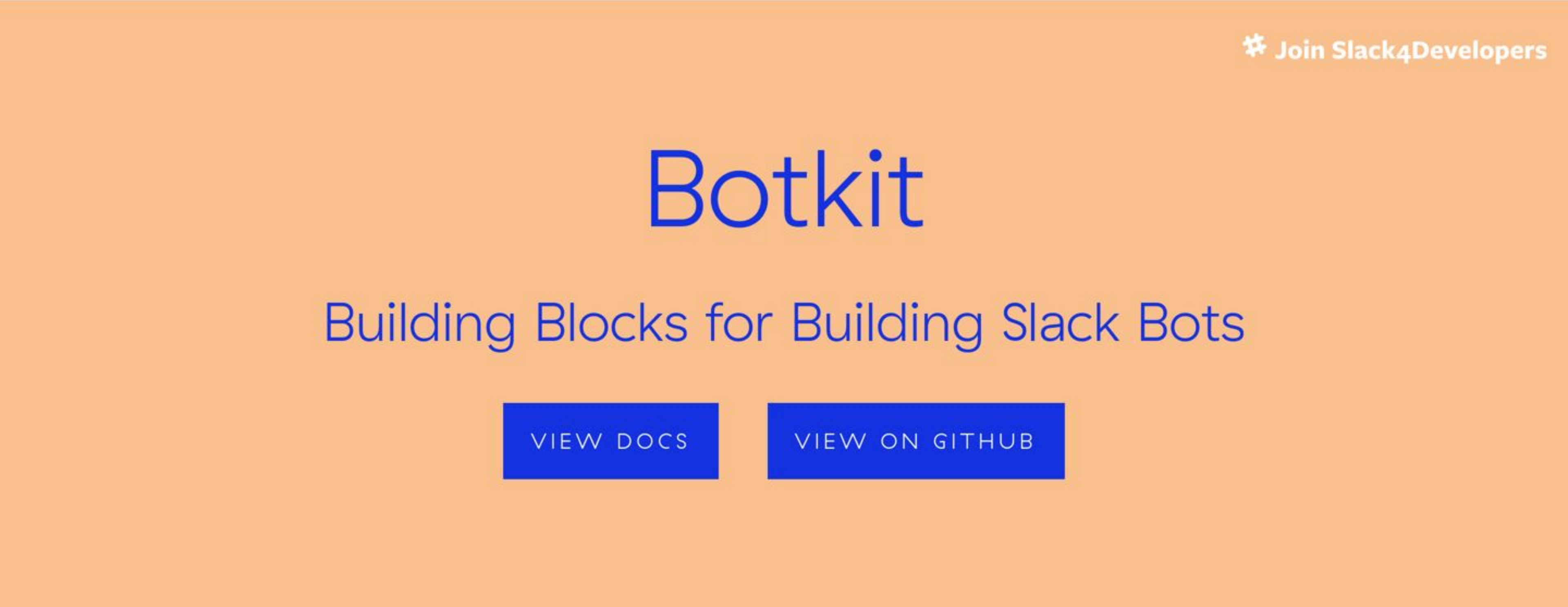
Nikabot
Hands-free team tracking and automatic project reports

Workbot
Chat with Salesforce, Github, Mailchimp, Zendesk and 100 other apps.

Awesome
Stay in sync, cut through the noise, and uncover insights.

DBOT by Demisto
DBOT checks files, URLs and IPs on selected channels and warns you when you may b...

Botkit



Join Slack4Developers

Botkit

Building Blocks for Building Slack Bots

Botkit

Botkit is a Node.js based framework that makes it easier to build a bot. Integration with Slack is automatic. The code uses understandable metaphors like “hears,” “asks” as actions.

Botkit

Botkit is a Node.js based framework that makes it easier to build a bot. Integration with Slack is automatic. The code uses understandable metaphors like “hears,” “asks” as actions.

Pros: Small/clear codebase, open source, written in node, easy integration with Slack (and wit.ai), very friendly & helpful community (follow the author, @benbrown on twitter and medium), great docs, company based around the project (howdy.ai)

Botkit

Botkit is a Node.js based framework that makes it easier to build a bot. Integration with Slack is automatic. The code uses understandable metaphors like “hears,” “asks” as actions.

Pros: Small/clear codebase, open source, written in node, easy integration with Slack (and wit.ai), very friendly & helpful community (follow the author, @benbrown on twitter and medium), great docs, company based around the project (howdy.ai)

Cons: ES5/callbacks (vs ES6/promises), competing (to some degree) with Microsoft (just announced a bot building framework)

3 Botkit Projects

1. Git clone Botkit, and fire up the main file, bot.js
   a. `git clone git@github.com:howdyai/botkit.git`
   b. `cd botkit && npm install`
   c. Get a Bot token from Slack:
      - http://my.slack.com/services/new/bot
   d. Run your bot from the command line:
      - `token=<MY TOKEN> node bot.js`
   e. Try to read and change the code and see what happens. My first bot using Botkit was called shithead, and all he did was add negativity.

3 Botkit Projects

2. Somerset

A summarizing Slack bot, for when you read too much for your own good and want everyone else to get on your level.

Building Somerset

3 Botkit Projects

3. Advanced

A bot that trains itself

Make a Self-Training, NLP-Driven Slack Bot

DEPENDENCIES & SETUP

Node, npm, Slack

Dependencies

- Git
- Node/npm
- Slack

Git

Mac

Install Homebrew (one-liner), then

```
brew update
brew doctor
brew install git
```

Linux

```
sudo apt-get install git
sudo yum install git
```

Depending on flavor of linux

node.js

Node is amazing. If all you get out of this is starting to play with node, I am happy.

- **Node is JavaScript, but outside the browser.** There are other runtimes for JS outside the server: Rhino, Chakra, Nashorn, SpiderMonkey, but none of them have anything near the adoption of node.
- The JS event loop forces you to think asynchronously. Who cares what you *can* do in a language - what does it *force* you to do? Because that's what you will actually do.
    - Time spent learning the hard parts of Javascript is not time wasted. You need to really understand JS to use it effectively. Here is an interactive visualization for understanding the event loop.

npm

Node comes with the node package manager, npm. Most programmers use Javascript, so there are A LOT of packages that are only an `npm install package` away. There are problems with npm and overusing it, but it's fantastic when you realize someone already wrote a package that you were going to spend a week working on.

node/npm

Getting it is easy, just visit the node homepage.

CONTACT

Takeaways, and Thank You

Stay in Touch

Open Source
I am going to be open-sourcing a few projects soon, including ResumeBot. Currently, one project I have that you might be interested in is a generator for a static site with a resume.

Follow
I want to see what you build! Also, if you want updates on things like this, interesting articles, and job opportunities, follow me on twitter @sam_havens. You can find my Medium account, LinkedIn, GitHub, etc on my website (made with the generator mentioned above) at samuelhavens.com

Final Remarks

Bots are cool
They are fun to make, they are an interesting blend of new and old tech, and they have a different set of problems than other applications

You are (most likely) young
I didn’t really start programming until I was 27. I know there are some kids who write their first compiler at 7 or whatever, but just because that isn’t you doesn’t mean there isn’t space for you in the community

Thank You

Thanks to all of you for having me, Ian for inviting me, and CarLabs for letting me take the day off for this!

Q&A

BUILD

Let’s Make Something

Slides and tutorial

This deck is available online: samuelhavens.com/botkit_talk.pdf

Some tutorials leading you through building a bot for Slack with Botkit are on slide 32, right before the “Dependencies and Setup” slide.