Lesson 5

(Project 1: Making a bot for Slack)

Learning Objectives

- Install and configure all utilities needed to run a Hubot
- Write scripts that allow our Hubot to interact with users in Slack

Last week…

- Functions: how to write them, call them, pass them arguments, and return values from them

- Refactored code to make it cleaner and easier to read

Project 1: Hubots

- We'll build our own chat bots that will interact with real users in Slack
- We'll use Hubot, a framework for creating chat bots
- We'll write the code for all our chat bots in JavaScript

Some vocab

- **chat bot:** a program that is programmed to interact with users as if it were a person. (You've probably seen these in Twitter, etc.)
    - Called "bot users" in Slack, but usually just "bots" for short.

- **Slackbot:** This is the one builtin bot in Slack. It's the bot that welcomes you when you join and tells you when someone you just messaged is probably asleep.

More vocab

- **Hubot:** a framework made by GitHub, to speed up the process of developing bots on many platforms, including Slack.
    - The bots we create with it are also called Hubots.
    - Runs on Node.js
- **Yeoman:** a set of tools that provides a scaffolding for getting web apps up and running quickly.
    - Takes care of loading necessary libraries, setting up our Hubot, etc. Does all the grunt work so we can write code

Last of vocab

- **CoffeeScript:** A variant of JavaScript
  - Some people prefer to write their JavaScript in CoffeeScript because they feel it's easier to read
  - Needs to be translated into JavaScript in order to run
  - CoffeeScript often associated with Hubot
  - You won't need to know CoffeeScript for this class!
  - If you find Hubot examples in CoffeeScript on the Internet, you can translate them at js2.coffee or coffeescript.org

The lab

- We've sent each person a Slack API token, which you'll need to follow the instructions to run Hubot.

- You've also all received invitations to join two other Slack teams (called "Team 2 Hubots" and "Team 3 Hubots"), because Slack only allows 10 bots per team, so we need 3 teams for this lab.

- If your bot is configured to be in one of the new teams, I've already sent you a message telling you so, and you have to join at least that team.

- We'll go through the Hubot installation together. Instructions at https://gist.github.com/richardharrington/29d20de8f62e0371f37b2ac3d0308885

Codealong: hubots

- If you wish you may refer to this page as we go along or after, which contains much of this material:

- http://bit.ly/jsdev2-lesson5-hubot-intro

Exercise: **.hear** and **.send**

- Make your bot listen for anyone saying the word “dog”

- Broadcast the message of your choice to the channel.

- How will you test that this works?

Exercise: **.respond**

- Make your bot listen for anyone addressing it directly (using **.respond**) with the question “What’s up?”

- Have your bot broadcast to the channel (using **.send**) with the message of your choice.

- How will you test that this works?

- **Bonus:** Make your bot broadcast to the channel by saying “You already asked me that **n** times!”, with n being how many times the question has been asked.

Exercise: **.reply**

- Make your bot listen for anyone saying the word “cat” (using **.hear**) and

- Have it broadcast “nice kitty” to the entire channel (using **.send**).

- Also have your bot listen for anyone saying the word “mountain lion” (using **.hear**),

- Have it respond DIRECTLY to the person who spoke with an @-mention (**.reply**), saying something like, “Shh! Don’t talk about those creatures.”

Exercise: **.random**

- Make an array of words, on some topic. (Like fruits, for example…)
- Have your bot listen for anyone addressing it directly with the words “tell me” (using **.respond**,
- Broadcast to the channel (using **.send**), “Tell you what? I don’t know anything about <x>!” where x is a random word from your array of words.

Exercise: wild cards

- Make your bot listen for someone addressing it directly (using **.respond**) with “My pet is a mountain lion” or “My pet is a cat”

- Depending on which kind of cat gets mentioned, broadcast two different messages.

- Have it broadcast a third message if the person says “My pet is a <something else>”

- **Bonus:** Do something similar, but using two wildcards in one input. Hint: **res.match[2]** will give you the value of the second wildcard.

Exercise: using two wildcards

- Using **parseInt**, When your bot is addressed directly (using **.respond**) with the arithmetic question “What is <x> plus <y>”, where x and y are whole numbers, have it respond directly (with **.**reply) with “<x> plus <y> is <sum of x and y>”.

- Example:

  richardh @bot: What is 6 plus 10

  bot @richardh: 6 plus 10 is 16

Final practice

- Play around with all this stuff. See if you can make something we haven't seen. Or just get some practice.

- **Bonus**: write some code for your bot to enable it to convert dollars to bitcoin (today's conversion rate is $463.83 per bitcoin). Example:

    richardh convert $1000 to bitcoin

    bot 2.155962313778 76 bitcoin

You will need to look up what **parseInt** does

Homework

1. Write a function that does what the Hubot **.random** function does: it takes an array as input, and returns a random element from the array.
2. Solve the first two problems on https://projecteuler.net/

—

- show the answer and the code you used to calculate it.
- Use what you need from what we've learned so far!

3. **Bonus:** solve the third problem on Project Euler as well.