

Forum:[Hack Forums](#)

Thread: [Custom Slack Bot -- Parsing Commands](#)

Ichii - April 24, 2017, 9:21 am

I am writing a custom slack bot that will handle business tasks as well as some fun features. The way I currently have the bot handle incoming commands "works", but I want it to be cleaner.

The way I have it now is one ugly if/else block statement which can be found below:

PHP Code

```
<?php
```

```
def handle_cmd(command, channel):
    """ Handle all valid and invalid commands """
    resp = ""
    cmd = command.lower()

    if cmd == 'help':
        resp = botcommands.help()
    elif cmd == 'get time':
        resp = botcommands.get_time()
    elif cmd == 'ping':
        resp = botcommands.ping()
    elif cmd == 'tell me a joke':
        resp = botcommands.joke()
    elif cmd[0:5] == 'roast':
        resp = botcommands.insult(cmd[6:])
    elif cmd[0:4] == 'echo':
        resp = botcommands.echo(cmd[5:])
    elif cmd[0:10] == 'youtube me':
        resp = botcommands.youtube.search(cmd[11:])

    else:
        resp = "I'm not sure what you're asking?"
        sc.api_call("chat.postMessage", channel=channel,
                    text=resp, as_user=True)

    sc.api_call("chat.postMessage", link_names=True, channel=channel,
                text=resp, as_user=True)
```

What I want to move to is a dictionary based approach. Valid trigger commands as keys and the corresponding methods as values. See below:

PHP Code

```
<?php
```

```
def handle_cmd(command, channel):
    """ Handle all valid and invalid commands """
    resp = ""
    cmd = command.lower()

    APPROVED_CMDS = {'tell me a joke': botcommands.joke(),
```

```

'ping': botcommands.ping(),
'help': botcommands.help(),
'time': botcommands.get_time(),
'roast': botcommands.insult(cmd[6:]),
'echo': botcommands.echo(cmd[5:]),
'youtube me': botcommands.youtube.search(cmd[11:])}

```

```

if cmd in APPROVED_CMDS.keys():
    // activate corresponding method
else:
    // do nothing

```

```

sc.api_call("chat.postMessage", link_names=True, channel=channel,
            text=resp, as_user=True)

```

The issue I am running into is that all commands come in on one line, like this:

PHP Code

```
<?php
```

```
@bot youtube me <query>
```

PHP Code

```
<?php
```

```
@bot roast <person>
```

PHP Code

```
<?php
```

```
@bot help
```

Not all trigger commands are the same length and not all trigger commands have additional parameters, which is why my current approach is to slice the string and pass in the correct parts of the whole string. This is sloppy and I know there's a better way to do it, just struggling to find that correct way. Not sure how to make the bot know how to differentiate between what is a trigger command and what is a parameter, other than slicing the string.

Open to suggestions

magnuscarter - April 24, 2017, 10:12 am

Nice code, I've been parsing data from graph databases into slack, and this could come in handy for controlling the business.

(???)?(???? - April 24, 2017, 11:09 am

I may be misunderstanding, but in your dictionary, put the values as this:

```

Code {
"key": func
}

```

And then whenever you get it, call the function directly. Right now, from what I'm understanding, is the function is getting called whenever you put it in the dictionary. You need to store a reference to the function itself.

ichi - April 25, 2017, 8:13 am

(Apr 24, 2017 11:09 AM)(???)?(???? Wrote: [\[CITE\]](#) I may be misunderstanding, but in your dictionary, put the values as this:

```

Code {
"key": func
}

```

And then whenever you get it, call the function directly. Right now, from what I'm understanding, is the function is getting called whenever you put it in the dictionary. You need to store a reference to the function itself.

Well the functions are already in the dictionary, waiting to be called once user input gets mapped to its corresponding key.

This issue I'm having is actually making the program understand what is the trigger command (help, roast, youtube me) and what is the parameter to pass into the function. As it currently stands, I'm splitting the string by length and passing it in that way; but some commands have extra parameters and some don't, I.E. the youtube search feature versus the help feature.

(?°?)?(???? - April 25, 2017, 12:23 pm

(Apr 25, 2017 08:13 AM)Ichi Wrote: [\[CITE\]](#) Well the functions are already in the dictionary, waiting to be called once user input gets mapped to its corresponding key.

This issue I'm having is actually making the program understand what is the trigger command (help, roast, youtube me) and what is the parameter to pass into the function. As it currently stands, I'm splitting the string by length and passing it in that way; but some commands have extra parameters and some don't, I.E. the youtube search feature versus the help feature.

One of us is confused. Maybe this will help explain my point:

```
Code def foo():  
    print("foo")
```

```
def bar():  
    print("bar")
```

```
d = {  
    "foo": foo(),  
    "bar": bar()  
}
```

Run that - you'll notice that it prints "foo" and "bar". It's storing the results of the function in your dictionary, not the function itself. To fix:

```
Code d = {  
    "foo": foo,  
    "bar": bar  
}
```

```
d["foo"]()
```