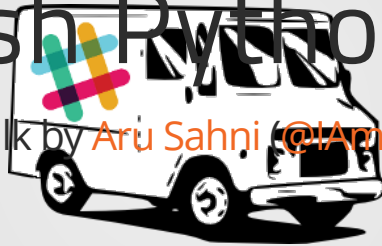


# Food Truck Tracking in Slack

Providing Prescient Postings for  
Peckish Pythonistas

A talk by Aru Sahni (@IAmAru)



# This talk...

- Won't cover: Aggregating truck locations.
- Won't cover: Parsing locations from Tweets using NLP.
- Won't cover: Using your collection of James Bond gadgets to affix small transponders to moving vehicles at dangerously high speeds.
- Will cover: Using Python to connect existing services.

# Problem

1. Developers need food.
2. Food comes from trucks.
3. Truck info is poorly aggregated.

# Solution?

Food truck tracking sites!

# Problem

- They suck.
- Many clicks, no deep linking.
- Coworkers spend a lot of time independently looking up listings.

# Slack

FooPlugins

CHANNELS

# affiliates

# blog-ideas

# blogs

# documentation

# general

# github

# helpscout

# helpscout-general

# plugin-ideas

# random

# redesign

# releases

# security

# seo-commentary

# trello

Create a channel...

DIRECT MESSAGES

slackbot

bradvin

mattcromwell

steveush

PRIVATE GROUPS

adam

online

#blogs

Search


?

≡

🔖

3

FooPlugins » Blog 3:09 PM

 **FooBox v2.3 – The Best Mobile Responsive Lightbox Ever Conceived by Man**

We've just pushed an update to FooBox that brings it to v2.3.0.4.1. It's now available as an automatic update for current license holders. This is a massive update, with a lot of important things to report. You can read the entire Changelog at the end of this post, but we'll list some of the most interesting [...]

The post FooBox v2.3 – The Best Mobile Responsive Lightbox Ever Conceived by Man appeared first on FooPlugins.

FooPlugins

**FooBox Version 2.3 Released**

We've just pushed an update to FooBox that brings it to v2.3.0.4.1. It's now available as an automatic update for current license holders. This is a massive update, with a lot of important things to report. You can read the entire Changelog at the end of this post, but we'll list some of the most interesting [...]


Wed, May 7th at 10:57 AM

adam 4:48 PM

Interesting article with a couple social sharing icon removal cases

<http://feedly.com/e/ui7jEpOT>

It might seem foolish to even ask the question "Should I have social media sharing buttons on my site?" because the answer seems self-evident. I'm a firm belie



mattcromwell 5:05 PM

That's a great article. I posted in AWP and it's getting an interesting discussion going:

<https://www.facebook.com/groups/advancedwp/permalink/721537224575161/>

Facebook

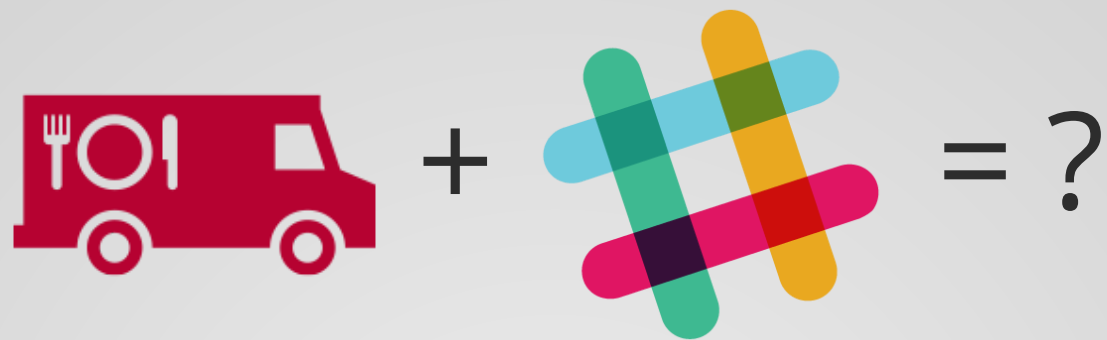
Facebook is a social utility that connects people with friends and others who work, study and live around them. People use Facebook to keep up with friends, upload an unlimited number of photos, post links and videos, and learn more about the people they meet.

👤

😊

# Slack

- Web-based messaging client.
- Similar to IRC w/channels.
- Collaborative features out of the box.
- **All our devs use it.**
- **Extensible.**





# A solution

1. The user asks Slack for nearby food trucks.
2. Slack queries a webservice for trucks in the area.
3. Slack publicly displays the results (so others don't have to ask).

# Listing food trucks

No formal API available.

No problem - scraping to the rescue.



PSA: Be respectful to the sites you scrape.

# Scraping with Python

- Heavy duty? **Scrapy**.
- Everything else: Roll your own.

# A custom scraper

- Requests
- BeautifulSoup

# Your first request

```
import requests
```

```
url = "http://arusahni.net/slack-demo/listing.html"  
response = requests.get(url)
```

```
# Print the full page markup  
print(response.text)
```

# Output

```
<html>
<body>
<ul>
  <li>District Taco</li>
  <li>Pho-bachi</li>
  <li>Astro Fried Chicken and Doughnuts</li>
</ul>
</body>
</html>
```

# Getting fancy

```
from bs4 import BeautifulSoup

soup = BeautifulSoup(response.text)
trucks = soup.find_all("li")

for truck in trucks:
    print(truck.text)
```



# Output

District Taco

Pho-batchi

Astro Fried Chicken and Doughnuts

# Putting it all together

```
import requests
from bs4 import BeautifulSoup

URL = "http://arusahni.net/slack-demo/listing.html"

def scrape_food_trucks():
    response = requests.get(URL)
    soup = BeautifulSoup(response.text)
    trucks = soup.find_all("li")
    return [truck.text for truck in trucks]
```

# Slacking off

Integrating with Slack is straightforward...

1. Slash-command calls the food truck script.
  - *e.g. /foodtrucks*
2. The script pipes output to Slack.

# Putting the script online

To be reachable, the script needs to live online.

- Flask
- Heroku

# Creating a webservice: Boilerplate

```
from flask import Flask

APP = Flask(__name__)

def scrape_food_trucks():
    # ...

@APP.route("/trucks")
def get_trucks():
    return ", ".join(scrape_food_trucks())
```

## Integration Settings

### Command

/foodtrucks

Examples: /deploy, /ack, /weather Reserved: /help, /me, /remind, /task, /bug, /feed, /shrug, /halp, /topic, /status, /away, /dm, /msg, /open, /join, /close, /leave, /mute, /who, /archive, /unarchive, /feedback, /hangout, /color, /colour, /invite, /commands, /play, /active, /part

### URL

The URL to request when the slash command is run.

### Method

GET

Should we do a POST or GET to the above URLs?

### Token

This token will be sent in the outgoing payload. You can use it to verify the request came from your Slack team.

vPpj7lO3xdOMK18EDDQWR5Bi

[Regenerate](#)

# Slash-command Payload

```
token = vPpj7l03xd0MK18EDDQWR5Bi  
team_id = T0001  
channel_id = C2147483705  
channel_name = <name>  
user_id = <user_id>  
user_name = <user_name>  
command = /foodtrucks  
text = <optional parameters>
```

# Protect ya neck

```
from flask import Flask, request, abort
APP = Flask(__name__)

TOKEN = "vPpj7l03xd0MK18EDDQWR5Bi"

@APP.route("/trucks")
def get_trucks():
    if request.args.get("token") != TOKEN:
        abort(401) # "Unauthorized" HTTP status code
    return ", ".join(scrape_food_trucks())
```



# Integrating with Heroku

- Hosted SaaS platform.
- Allows you to scale up and drop in new components.
- Uses git for code deployment

# Add pre-requisites

```
# Install server
```

```
pip install gunicorn
```

```
# Capture dependencies
```

```
pip freeze > requirements.txt
```

```
# Specify Python version
```

```
echo "python-3.4.1" > runtime.txt
```

```
# Point the server to the app
```

```
echo "web: gunicorn foodtrucks:APP --log-file -" > Procfile
```

# Get things going

```
# Instantiate a Heroku project and tie it to your repo  
heroku create --http-git
```

```
# Push your master branch to Heroku  
git push heroku master
```

```
# Get APP URL from above and enter it in the Slack configuration
```

# Give it a whirl!



**slackbot** 4:53 PM Only you can see this message

*District Taco, Pho-bachi, Astro Fried Chicken and Doughnuts*

# Kinda sorta done

- We've got listings!
- ... but I'm the only one that can see them!

# Incoming webhooks

- Allow external services to push data into Slack.
- Deliver JSON payloads to predetermined URIs.
- Can be used to broadcast messages to a room.

# Appearance

## Webhook URL

Send your JSON payloads to this URL.

[Show setup instructions](#)

`https://hooks.slack.com/services/T03393R49/B033DG874/PCnkHoqG6XmP3c`

[Copy URL](#) • [Regenerate](#)

## Descriptive Label

Use this label to provide extra context in your list of integrations (optional).

Optional description of this integration

## Customize Name

Choose the username that this integration will post as.

Slacktrux

## Customize Icon

Change the icon that is used for messages from this integration.



[Change Icon](#)

# The payload

```
payload = {"text": ', '.join(scrape_food_trucks())}
```



# Updating our service

```
import json
import requests

TOKEN = "vPpj7l03xd0MK18EDDQWR5Bi"
WEBHOOK_URL = "https://hooks.slack.com/services/T03...6XmP3cY6J95xt95w"

@APP.route("/trucks")
def get_trucks():
    if request.args.get("token") != TOKEN:
        abort(401) # unauthorized
    payload = {"text": ', '.join(scrape_food_trucks())}
    resp = requests.post(WEBHOOK_URL, data={"payload": json.dumps(payload)})
    if resp.status_code != 200:
        return "HTTP {}: {}".format(resp.status_code, resp.text), 500
    return ""
```

# And that's it!



**Slacktrux** 10:23 PM

District Taco, Pho-bachi, Astro Fried Chicken and Doughnuts

# What's next?

- Cache food truck locations.
- Move scraping off the main webserver thread.
- Replace the food truck tracking site.
- Have the bot post whenever the tracking service updates the listings for your area.
- Start packing your lunch.

# Fin.

Aru Sahni (@IAmAru)

# HUMANgEO

We're always looking for talented Python developers.