Analyze This!

Simple Calculator Applications Using Bottle.py and Goggle App Engine

John Hackett
jhackettga@gmail.com

Kids Had Some Fun This Summer…

John H. Hackett
August 27, 2011 via mobile

Our latest venture. Seeking first mover advantage in the lemonade space for [redacted]. Initial proposed price points were $20 and $10... (they settled on 25 cents)

The Lemonade Stand Revenue Estimator

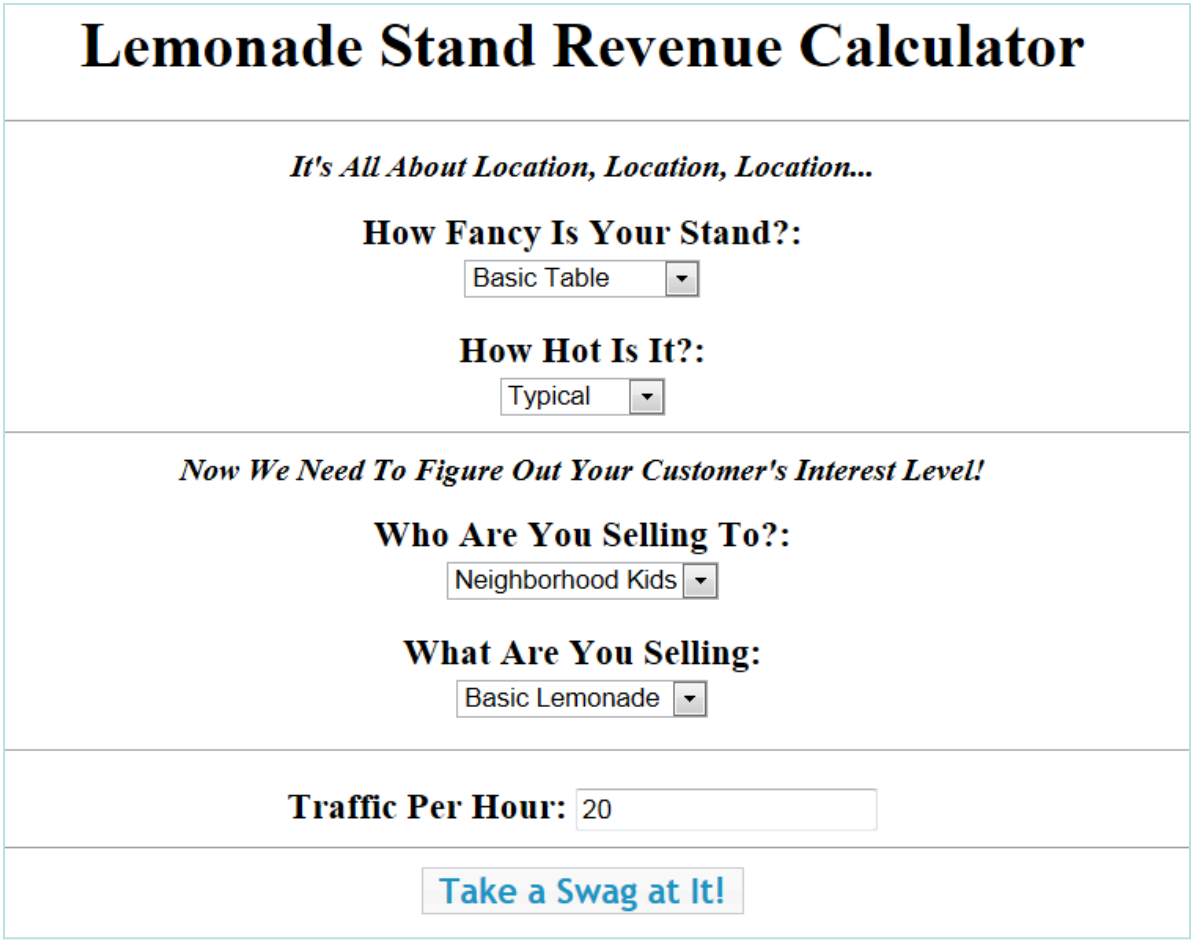
A pack of half-baked assumptions masquerading as serious analytics…

```
def lemonade_revenue(params):
        """Accepts list of form parameters as a dict; returns string with estimated sales per hour"""

        # remove junk values from the list of parameters
        params = dict([(k,v) for k,v in params.items() if len(v)])

        # convert form inputs into model
        presence = presence_value.get(params.get('sitetype',1))
        day = day_value.get(params.get('interestlevel',1))
        audience = audience_value.get(params.get('target_audience',1))
        revenuesource = revenuesource_value.get(params.get('revenuesource',1))

        # marketing math
        conversion_rate = min([.25*presence*day,.7])
        sale_value = min([.50*audience*revenuesource, 5])
        predicted_revenue =  float(params.get('visitors',0)) * conversion_rate * sale_value

        return "We Predict Your Stand Will Make $%s Per Hour" % int(roundoff(predicted_revenue))

# unpack information from the request
sitetype_map={'1':'Basic Table','2':'Big Sign', '3':'Booth', '4':'Dancing Person'}
revenuesource_map={'1':'Basic Lemonade','2':'Fancy Lemonade','3':'Drinks & Snacks','4':'Red Bull'}
target_audience_map = {'1':'Neighborhood Kids','2':'Soccer Moms', '3':'Tri-Atheletes','4':'Entrepreneurs'}
interestlevel_map = {'1':'Freezing','2':'Typical', '3':'Hot', '4':'Scorching'}

# analytics parameters
presence_value = {'1':1,'2':1.3, '3':2,'4':4}
day_value = {'1':.3,'2':.8, '3':1.2,'4':3}
audience_value = {'1':.5,'2':1, '3':2,'4':4}
revenuesource_value = {'1':1,'2':1.3, '3':2,'4':3}
```

Insert Your Own Secret Formula Here…

Math expressed in < 20 lines of Python

The Front End (bottledemo.appspot.com)



HTML Form Dressed Up With Jquery / ThemeRoller

Why This Matters…

- How Analytics Projects Work In Practice:
  - Lots of crunching to derive “the formula”
  - Use Proprietary Data / Confidential Policy Decisions
  - Then Need To Share Insights
- However, analytics insights are often:
  - Time Consuming to Communicate
  - Impossible to Understand
  - Trivial to Copy
- Simple web-apps like this allow you to:
  - Simplify / share insights with a large audience
  - Without exposing the formula / data

Implementation: Server Side (bottle.py)

```
import bottle
from bottle import route, run, view, static_file, error, validate, app, redirect, request, template

def main():
        bottle.run(server='gae')

@route('/',  method=['GET'])
def serve_initial_revenue():
        return template("templates/%s.tpl" % 'lemonade-stand-revenue-calculator',
            sitetype='Basic Table',
            interestlevel='Typical',
            target_audience='Neighborhood Kids',
            revenuesource='Basic Lemonade',
            visitors="value='20'",
            answer="Enter Some Assumptions and Let's Get Started!"
            )

@route('/', method=['POST'])
def serve_calc_revenue():
        return template("templates/%s.tpl" % 'lemonade-stand-revenue-calculator',
            sitetype=sitetype_map[request.forms.get('sitetype','1')],
            revenuesource=revenuesource_map[request.forms.get('revenuesource','1')],
            interestlevel=interestlevel_map[request.forms.get('interestlevel','1')],
            target_audience=target_audience_map[request.forms.get('target_audience','1')],
            visitors="value='%s'" % request.forms.get('visitors',25000),
            answer=lemonade_revenue(dict(request.forms))
            )


@route('/static/:name')
def serve_static(name):
    return static_file(name,root="static")

@error(403)
@error(404)
def mistake_403(code):
    return "<h2>I'm Sorry Dave, I'm afraid I can't do that....</h2>"
```

Build Web Page
In Your Choice of
Template Languages

GET => Defaults
POST => Calc Value

Why Google App Engine?

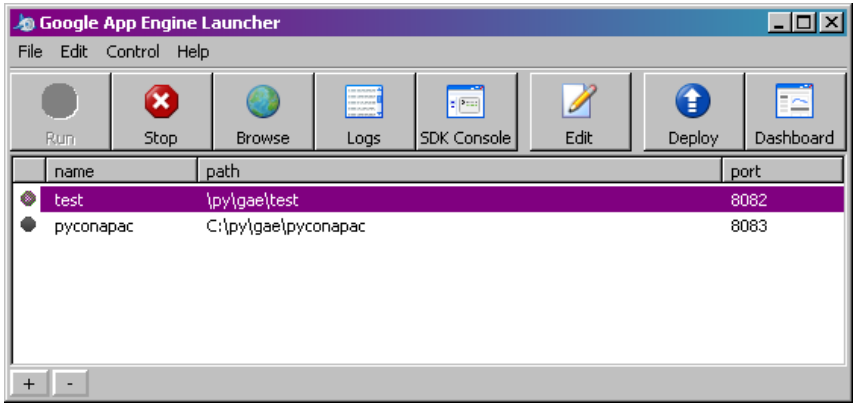
- Small Projects: Effectively Free
  - Subdomain on appspot.com (10 apps / account)
  - Free up to a monthly limit (pageviews / storage)
- Minimal System Administration Required
  - Don’t need to configure / manage server
  - Apps run in a sandbox environment
  - Scale automatically
- Google Integration / 3rd Party Libraries:
  - Google Accounts => User Management
  - Google API’s/Libraries: Search, Mail, Map Reduce, etc.
  - 3rd Party: Django, Numpy, PIL, lxml, jinja (templates)

Potential Constraints

- Sandbox Limitations
  - Connectivity: Web Requests / Email Only
  - Persistence:
    - Cannot write the file system
    - Can only read a file if you uploaded it with code
    - Must use Google's Datastore / Memecache Options
  - Code Execution:
    - Only if triggered by web request or queued/scheduled task
    - Must return data within 60 seconds
- Strategic Implications
  - Cannot use many popular databases and Python modules
  - Not necessarily be a cheaper option at scale
  - Google "lock-in": GAE modules / services

Moving Apps To GAE

Desktop Dev Environment



YAML Config File

```
application: bottledemo
version: 1
runtime: python
api_version: 1

handlers:
- url: /static
  static_dir: static

- url: /.*
  script: lemonade.py
```

Access Through Google Account

My Applications

‹ Prev 20 **1-1 of 1** Next 20 ›

Application	Title	Billing Administrator	Storage Scheme	Current Version
bottledemo	Sample Calculator Using Bottle		High Replication	1

Create Application

You have 9 applications remaining.

‹ Prev 20 **1-1 of 1** Next 20 ›

Want to Know More?

- Google App Engine
  - Main Site
  - Python Tutorial
- bottle.py web framework:
  - Project Site
  - Tutorials & Resources
- More Advanced Example of Calculator Concept
  - Calculator: Website Revenue Estimator
  - Underlying Analytics : Website Revenue Model Study