

Google App Engine in 3 Hours or less

Python in the "Cloud"



Part 1: Writing App Engine Applications

* Time: 45 Minutes

Hello World is easy

- * #1 Sign up: <http://code.google.com/appengine/>
- * Create two files a hello.py and app.yaml

Hello World is easy

app.yaml

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

```
handlers:  
- url: .*  
  script: hello.py
```

Hello World is easy

hello.py

```
import wsgiref.handlers
from google.appengine.ext import webapp

class MainHandler(webapp.RequestHandler):
    def get(self):
        self.response.out.write('Hello PyWorks')

def main():
    application = webapp.WSGIApplication([('/', MainHandler)],
                                         debug=True)
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
```

Tutorial URL

* <http://row.appspot.com>

Positives

- * Python is a great language
- * Rich APIs: GData, User's API, AJAX, https
- * Free to start
- * Simple to use
- * Scales Up, Medium, and small

Negatives

- * Still in beta
- * Commercial version not available yet...soon
- * Many features not available yet...soon
- * Proprietary database, not SQL

Live Sample: Greedy Coin and row

* Simple Coin Changer: <http://greedycoin.appspot.com/>

Part 2: Using GData

* Time: 45 Minutes

GDATA

- * Independent of APP Engine
- * Calendar, Spreadsheet, Excel, etc.
- * A good ecosystem for developers on App engine
- * GData is a manifestation of Google's goal to organize the worlds data.

GDATA Programmatic Login

* LAB: Follow Steps from download

GDATA AuthSub

* Demo/Code: <http://row.appspot.com>

Part 3: Google's AJAX APIs

* Time: 45 Minutes

Part 3: Google's AJAX APIs

- * Dynamic Feed Control
- * Look at `feed.html/feed.py`

Part 3: Google's AJAX Hosting

```
* <script src="http://www.google.com/jsapi"></script>
```

```
<script>  
  // Load jQuery  
  google.load("jquery", "1");  
  // Do jQuery  
</script>
```

Questions?