



# Hands-On with *Google App Engine* workshop

**Ikai Lan**  
**Developer Relations**

Google App Engine

[ikai.l@google.com](mailto:ikai.l@google.com)

OSCON 2010

[bit.ly/gcodelabs](http://bit.ly/gcodelabs)



# Your Workshop Instructor (drill sgt)

- **Engineer**

- Lots of experience building and scaling web applications (Java/Ruby on MySQL)
- Front-end - JavaScript, OpenSocial
- Focused on teaching and supporting Google App Engine

- **Contact**

- Twitter: @ikai
- Email: [ikai.l@google.com](mailto:ikai.l@google.com)

# Google App Engine code lab



- **Objective**
  - Hands-on experience developing\* an App Engine app
- **Requirements**
  - Computer, Python 2.5.x & App Engine SDK
    - Text editor and command shell or IDE
  - Optional: valid Google account, SMS on cell
- **Session**
  - Single app, seven fast iterations. Crash? Go to teammate!
  - Copy, edit, run (lather, rinse, repeat)
  - Look for diffs/changes in pink

# Check your Python install



Linux/Mac

\$ python

In Windows:

- Go to Start->Programs->Python

**Make sure you are running Python 2.5!**

# Download!

- SDK

- <http://code.google.com/appengine/>

- Codelab

- <http://bit.ly/gcodelabs>



# Why App Engine?



- Easy to start
  - Download the SDK and begin writing code!
- Easy to scale
  - Push your code onto Google servers, capacity added as needed

# Tips before we begin



- Be mindful of whitespace
- Follow along at <http://bit.ly.godelabs>
- Ask questions

# Create your directory

- **YOUR\_APP\_DIRECTORY/**
  - **main.py**
  - **app.yaml**
  - **index.yaml**



# Set up your config (app.yaml)

**application: YOUR\_APP\_ID**

**version: 1**

**runtime: python**

**api\_version: 1**

**handlers:**

**- url: /\***

**script: main.py**

# Your first App Engine App (main.py)

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp import util
```

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

```
def main():
    application = webapp.WSGIApplication([('/',
        MainHandler)], debug=True)
    util.run_wsgi_app(application)
```

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

# Run your app!

From the launcher click "Run". OR

Command line:

```
$ dev_appserver.py YOUR_APP_DIR
```

Browse to: <http://localhost:8080>

Alternatively: <http://127.0.0.1:8080>

# Putting code in a function (main1.py)

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
```

```
class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.out.write('Hello World!')
```

```
application = webapp.WSGIApplication([
    ('/', MainPage),
], debug=True)
```

```
def main():
    run_wsgi_app(application)
```

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

# Add HTML (main2.py)

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
```

```
class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/html'
        self.response.out.write('<h1>Hello World!</h1>')
```

```
:
```

# Add HTML form (make3.py)

```
class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.out.write('<h1>Hello world!</h1>')
        self.response.out.write("""
            <form action="/sign" method=post>
            <input type=text name=content>
            <input type=submit value="Sign Guestbook">
            </form>
        """)
```

# Add signing handler (main4.py)

```
from google.appengine.ext import webapp
:
class MainPage(webapp.RequestHandler):
:

class GuestBook(webapp.RequestHandler):
    def post(self):
        self.response.out.write(
            '<h3>You wrote:</h3>%s' % self.request.get('content'))

application = webapp.WSGIApplication([
    ('/', MainPage),
    ('/sign', GuestBook),
], debug=True)
```

# Storing data (main5.py)

```
:
from google.appengine.ext import db, webapp
:
class Greeting(db.Model):
    content = db.StringProperty(multiline=True)
    date = db.DateTimeProperty(auto_now_add=True)

class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.out.write('<h1>My Guestbook</h1><ol>')
        greetings = Greeting.all()
        for greeting in greetings:
            self.response.out.write('<li> %s </li>' % greeting.content)
            self.response.out.write("</ol><hr>
                <form action='/sign' method=post>
                <textarea name=content rows=3 cols=60></textarea>
                <input type=submit value='Sign Guestbook'>
                </form>")

class GuestBook(webapp.RequestHandler):
    def post(self):
        greeting = Greeting()
        greeting.content = self.request.get('content')
        greeting.put()
        self.redirect('/')
```

# View your data in admin console

**Browse to:**

**[http://localhost:8080/\\_ah/admin](http://localhost:8080/_ah/admin)**

**Alternatively:**

**[http://127.0.0.1:8080/\\_ah/admin](http://127.0.0.1:8080/_ah/admin)**

# Adding users (& authors) (main6.py)

```
:
from google.appengine.api import users
:
class Greeting(db.Model):
    author = db.UserProperty()
    content = db.StringProperty(multiline=True)
    date = db.DateTimeProperty(auto_now_add=True)

class MainPage(webapp.RequestHandler):
    def get(self):
        user = users.get_current_user()
        if user:
            self.response.out.write('Hello %s!' % user.nickname())
        else:
            self.redirect(users.create_login_url(self.request.uri))
            self.response.out.write('<h1>My Guestbook</h1><ol>')
:
class GuestBook(webapp.RequestHandler):
    def post(self):
        greeting = Greeting()
        user = users.get_current_user()
        if user:
            greeting.author = user
            greeting.content = self.request.get('content')
```

# Adding an HTML template (main7.py)

```
:  
from os import path  
:  
from google.appengine.ext.webapp.template import render  
:  
  
class MainPage(webapp.RequestHandler):  
    def get(self):  
        user = users.get_current_user()  
        greetings = Greeting.all()  
        context = {  
            'user': user,  
            'greetings': greetings,  
            'login': users.create_login_url(self.request.uri),  
            'logout': users.create_logout_url(self.request.uri),  
        }  
        tpl = path.join(path.dirname(__file__), 'index.html')  
        self.response.out.write(render(tmpl, context))
```

# Adding an HTML template (index.html)

```
<html><body>Hello
{% if user %}
{{ user.nickname }}!
[<a href="{{ logout }}"><b>sign out</b></a>]
{% else %}
World!
[<a href="{{ login }}"><b>sign in</b></a>]
{% endif %}

<h1>My Guestbook</h1><ol>
{% for greeting in greetings %}
<i>
{% if greeting.author %}
{{ greeting.author.nickname }}
{% else %}
<i>anonymous</i>
{% endif %}
{{ greeting.content|escape }}
{% endfor %}
</ol><hr>
<form action="/sign" method=post>
<textarea name=content rows=3 cols=60></textarea>
<input type=submit value="Sign Guestbook">
</form></body></html>
```

# Now what?

- Keep going! We'll be here to help  
<http://bit.ly/gcodelabs>
- Push your application live  
<http://appspot.com>
- Read more documentation  
<http://code.google.com/appengine>