Web Programming with Apache and Python

Balamurugan S
Email: balamurugan.sekar@gmail.com
Web: http://www.bitbala.org/
Agenda

- Introduction to Web
- Introduction to Apache
- Web Programming with Python and CGI
- Web Programming with Python and mod_python
Introduction to Web

- Internet
- Need for dynamic content over Web
- Web servers
Web servers

- Computer program which delivers the content to users over Hypertext Transfer protocol (HTTP) over the World Wide Web (WWW)
- Apache httpd, Apache Tomcat, Microsoft IIS, Google GWS, lighthttpd
How web server works?

1. Sends request
2. Request External executions
3. Sends response

Web Clients - Browsers
Web Servers
Applications
Apache - Installation

- Debian based systems
  apt-get install apache2
- RPM based systems
  yum install apache2
- Can be downloaded and compiled :)

Web Programming with Apache and Python
Introduction to CGI

- Common Gateway Interface
- Written using high level languages
- Commonly used High level languages are
  - C/C++
  - Python
  - Perl
  - PHP
- CGI is no longer widely used
- CGI is good for small applications
CGI

• How CGI scripts are different from normal scripts?
• What kind of Inputs it takes?
• How to generate the output?
  ▪ Since it is sending the data to Web browsers, it should send valid HTTP Data (MIME Headers + HTML)
Apache Configuration for CGI

- Change the content of /etc/apache2/sites-available

```xml
<Directory "/path/to/cgi/scripts">
  AllowOverride None
  Options +ExecCGI
  AddHandler cgi-script .py
  Order allow, deny
  Allow from all
</Directory>

<Directory "/path/to/cgi/scripts">
  Options All
</Directory>
```
Hello World!!!

- Helloworld.py

```python
# ! / usr / bin / python
print "Content-type: text/html\n\n"
print '<html>
print '<head>
print '<title>Hello World - First CGI Program</title>
print '</head>
print '<body>
print '<h2>Hello World! This is my first CGI program</h2>
print '</body>
print '</html>
```

Line 2 – Sends the HTTP Header to browser.
CGI Environment Variables

- CONTENT_TYPE, CONTENT_LENGTH, HTTP_COOKIE, HTTP_USER_AGENT, PATH_INFO, QUERY_STRING, REMOTE_ADDR, REMOTE_HOST, REQUEST_METHOD, SCRIPT_FILENAME, SCRIPT_NAME, SERVER_NAME, SERVER_SOFTWARE

To print the CGI Environment Variables

```python
#!/usr/bin/python

import os

print "Content-type: text/html\r\n\r\n";
print "<font size=+1>Environment</font>\r\n";
for param in os.environ.keys():
    print "<b>%s</b>: %s\r\n" % (param, os.environ[param])
```
cgi module

- Primary module to be used in CGI programming in python
- FieldStorage is the primary class that will be used
- A object has to be created for FieldStorage when python CGI scripts begins
- The object will read all the pertinent user info sent by web client in a dictionary
- Name given in the form will be keys and Values of the form elements will be values in dictionaries
Sending data over Forms - GET

userform_get.html

```html
<form action="user_get.py" method="get">
First Name: <input type="text" name="first_name">
<br />
Last Name: <input type="text" name="last_name" />
<input type="submit" value="Submit" />
</form>
```

user_get.py

```python
#!/usr/bin/python
import cgi, cgitb
user_data = cgi.FieldStorage()
first_name = user_data.getvalue('first_name')
last_name = user_data.getvalue('last_name')
print "Content-type: text/html\r\n\r"
print "<html>
print "<head>
print "<title>Sending Data - GET</title>
print "</head>
print "<body>
print "<h2>Hello %s %s</h2>" % (first_name, last_name)
print "</body>
print "</html>"
```
Sending data over Forms - POST

userform_post.html

```html
<form action="user_post.py" method="post">
First Name: <input type="text" name="first_name"/><br/>
Last Name: <input type="text" name="last_name"/>
<input type="submit" value="Submit"/>
</form>
```

user_post.py

```python
#!/usr/bin/python
import cgi, cgitb
user_data = cgi.FieldStorage()
first_name = user_data.getvalue('first_name')
last_name = user_data.getvalue('last_name')
print "Content-type: text/html\n\r\n"
print "<html>
print "<head>
print "<title>Sending Data - GET</title>"
print "</head>"
print "<body>"
print "<h2>Hello %s %s</h2>" %(first_name, last_name)
print "</body>"
print "</html>"
```
cgitb module

- This module provides a special exception handler for Python scripts
- This will help you in debugging the CGI scripts
- After activating this module, if an uncaught exception occurs, a detailed, formatted report will be displayed.
- To use the module you have to
  - import cgitb
- To enable the module
  - cgitb.enable()
Passing checkbox

userform_checkbox.html

```html
<form action="user_checkbox.py" method="POST" target="_blank">
<input type="checkbox" name="maths" value="on" /> Maths
<input type="checkbox" name="physics" value="on" /> Physics
<input type="submit" value="Select Subject" />
</form>
```

user_checkbox.py

```python
#!/usr/bin/python
import cgi, cgitb
form = cgi.FieldStorage()
if form.getvalue('maths'):
    math_flag = "ON"
else:
    math_flag = "OFF"
if form.getvalue('physics'):
    physics_flag = "ON"
else:
    physics_flag = "OFF"
print "Content-type: text/html"
print "<html><head><title>Checkbox</title>"
print "<body>"
print "<h2> CheckBox Maths is : %s</h2>" % math_flag
print "<h2> CheckBox Physics is : %s</h2>" % physics_flag
print "</body></html>"
```
Passing Radio buttons

userform_radiobutton.html

```html
<form action="radiobutton.py" method="post" target="_blank">
  <input type="radio" name="subject" value="maths" /> Maths
  <input type="radio" name="subject" value="physics" /> Physics
  <input type="submit" value="Select Subject" />
</form>
```

radiobutton.py

```python
#!/usr/bin/python

import cgi, cgitb

form = cgi.FieldStorage()

if form.getvalue('subject'):
    subject = form.getvalue('subject')
else:
    subject = "Not set"

print "Content-type: text/html\r\n\r\n"
print "<html><head><title>Radio</title>"
print "</head><body>"
print "<h2>Selected Subject is %s</h2>" % subject
print "</body></html>"
```
File Upload

fileupload.html

```html
<html>
<body>
<form enctype="multipart/form-data" action="save_file.py" method="post">
  <p>File: <input type="file" name="filename" /></p>
  <input type="submit" value="Upload" />
</form>
</body>
</html>
```

fileupload.py

```python
#!/usr/bin/python

import cgi, os
import cgitb; cgitb.enable()

form = cgi.FieldStorage()
filename = form['filename']

if filename:
    fn = os.path.basename(filename)
    open('/tmp/' + fn, 'wb').write(form[filename].read())
    message = 'The file "' + fn + '" was uploaded successfully'
else:
    message = 'No file was uploaded'

print 'Content-Type: text/html

<html><body><p>%s</p></body></html>' % (message,
```
mod_python

• A module to connect Python and Apache
• Main advantages of mod_python over traditional CGI is performance
• Sample performance data
  - Standard CGI: 23 requests/s
  - Mod_python cgi handler: 385 requests/s
  - Mod_python publisher: 476 requests/s

Tests are done with 1.2GHz Pentium machine sending 1000 requests with frequency of 1
mod_python - Installation

- Debian based Systems
  - `apt-get install libapache2-mod-python`
- RPM based systems
  - `apt-get install libapache2-mod-python`
mod_python - Configuring Apache

- Change the content of /etc/apache2/sites-available

```xml
<Directory "/path/to/scripts">
    SetHandler mod_python.py
    PythonHandler mod_python.publisher
    PythonDebug On
    PythonAutoReloading On
    PythonPath "sys.path + ['/path/to/modules']"
</Directory>
```
Hello World!!!

- **Helloworld.py**

```python
1 def index():
2    s = """\n3    <html>
4    <body>
5    <h2>Hello World! </h2>
6    </body>
7    </html>
8    """
9    return s
```

- URL to access this - http://localhost/hello.py/index or http://localhost/hello.py
Hello World!!! - one more time

- Helloworld_new.py

```python
s = """\n<html><body>
<h2>Hello %s!</h2>
</body></html>
"""

def index():
    return s % 'World'

def everybody():
    return s % 'everybody'

def _hide_this_function
    return "You cannot access this function"
```

- URL to access everybody function
  http://localhost/helloworld_new.py/everybody
- Function starting with _ will be hidden to the web
Sending form data

- Request object will hold the user data sent over Web

**user_form.html**

```html
<html>
<body>
<form action="show.py/output" method="post">
  <p>Type Your name:  <input type="text" name="username"></p>
  <input type="submit" value="Submit"></form>
</body>
</html>
```

**show.py**

```python
def output(req):
    name = req.form['username']
    s = "\n<html><body>
  <p>The submitted word was "%s"</p>
  <a href="user_form.html">Submit another word!</a>
</body></html>
"""
    return s % name
```

python
File Upload

fileupload.html

```html
<html>
<body>
    <form enctype="multipart/form-data" action="upload.py/upload" method="post">
        <p>File: <input type="file" name="filename" /></p>
        <p><input type="submit" value="Upload" /></p>
    </form>
</body>
</html>
```

upload.py

```python
def upload(req):
    filename = req.form['filename'].filename
    fp = '/tmp/' + filename
    f = file(fp, 'wb')
    file_contents = req.form['filename'].file
    data = file_contents.read()
    f.write(data)
    f.close

    return "File upload successful"
```
Thank you
References

• Python Reference - Dive into Python
(http://www.diveintopython.org)

• Web programming using CGI
(http://www.tutorialspoint.com/python/python_cgi_programming.htm)

• Web python tutorial -
(http://webpython.codepoint.net/)