Getting Started

James Gardner

January 20, 2005

http://www.pythonweb.org
docs at pythonweb.org

© 2002-2004 James Gardner All rights reserved.

Contents

1 Overview 1

2 Required Software 2

3 Quick Installation 2

4 Further Options 3

5 Testing the Examples 3

6 Where Next 3

Getting Started

1 Overview

```
"It is better to have a flexible module that can be used intuitively than
an all-singing, all-dancing framework that no-one can be bothered to learn."

- James Gardner
```

The Python Web Modules are a suite of simple and easy to use Python components desinged to allow developers to write Python CGI scripts or web applications with SQL databases, sessions and auth functions quickly and easily without having to install Web Application Servers.

A key part of the project is the creation of documentation and examples to allow developers with less time or expierince to understand the concepts of Python web programming without having to read all the sources and comments from the very beginning.

A *Beginner's Guide* to web programming with the modules is also available and there are a number of *examples*.

2 Required Software

To use all the features of the modules you will either need a binary distribution (available for Windows) or you will need to install the following software in addition to the source distribution of the web modules:

- Python Imaging Library http://www.pythonware.com/products/pil/
- MySQLdb Module http://sourceforge.net/projects/mysql-python
- SQLite http://www.sqlite.org/ and pysqlite http://pysqlite.sourceforge.net/
- libxml2, libxslt and the Python wrappers http://www.xmlsoft.org/

All downloads including Windows versions are available from http://www.pythonweb.org/projects/webmodules/.

Note: You can still use the web modules even if you do not have all the components listed above installed. The functionality will be reduced but the modules have been designed so that the templating, auth, session, mail, WSGI, webserver and databaase code will function with nothing more than a recent version of Python.

If for example you do not wish to use the MySQL database engine then you may choose not to install the `MySQLdb` module. If you do not install the Python Imagin Library you will not be able to create graphs and if you do not install libxml2 and libxslt you will not be able to do XSLT transforms.

3 Quick Installation

The Python Web Modules do not need formally installing although it is eaiest if you do install them. All the examples, tests and scripts will run straight from the distribution once it is unzipped (see note above about optional components).

If you wish to use the web modulest the only requirement is that Python can find the 'web' directory containg the web modules. This can be achieved in a number of ways:

1. **Easiest:** Execute the command below to install the modules to your Python 'Lib/site-packages' directory where Python looks for new modules:

   ```
   python setup.py install
   ```

2. Place the 'web' directory of the distribution in the same directory as the script which uses the modules since Python always searches the directory containing a script when looking for modules.
3. Modify your `PYTHONPATH` by:
   - Adding the following code in every file using the modules before you import the web modules:

     ```
     import sys
     sys.path.insert(0,'C:/path/to/distribution/directory')
     ```

     replacing 'C:/path/to/distribution/directory' with the path to the directory containing the 'web' directory.

     Note: You should not include the 'web' directory in the path otherwise Python will look for a directory called 'web' inside the 'web' directory and will not find it!
   - Alternatively, append the path to the directory containing the 'web' directory to the PYTHONPATH variable in the standard way for your operating system.

Once the web modules are imported, internal paths are setup so that you can import `Cheetah` or `pdbc` for example, but you must import the `web` module first.

4 Further Options

To compile all the Python files for faster execution you can run the 'compileAll.py' script in the 'scripts' directory. **Note:** Python automatically compiles files the first time it uses them anyway.

5 Testing the Examples

To get up and running with the examples run 'webserver.py' in the 'scripts' directory to start a webserver on port 8080. Load a web browser and visit http://localhost:8080/doc/html/web/examples.html and you should see the *examples page* of the documentation. You don't need to install the modules to run the tests, they will run from the unzipped source directory.

If you are using a UNIX operating system you will need to set the permissions of the files you wish to test so that the web server has permission to execute them. You can do this with the following command from a terminal:

```
> chmod -R 755 doc/src/lib
```

This will recursively change the permissions of the test scripts to allow the webserver to execute them.

Some of the tests use the Web Server Gateway Interface server. These can be tested by running 'WSGIserver.py' in the 'scripts' directory and visiting http://localhost:8000/doc/html/web/examples.html

6 Where Next

The next place to visit is the Beginner's Guide which will get you started programming CGI scripts and then web applications or the Module Reference which will give you detailed information on the API and way to use each of the modules or the Examples section to see the modules in action.