

Re: modpython, apache and windows

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-01/0753.html>

grahamd_at_dscpl.com.au

Date: 01/05/05

Date: 5 Jan 2005 14:25:28 -0800

Bob Van Zant wrote:

- > *Fortunately most of the Python-for-the-web implementations do not follow*
- > *closely to the PHP paradigm of web-based programming. There are some,*
- > *like PSP, that more closely model what PHP does.*
- >
- > *It is not uncommon to have something like index.py which does hardly*
- > *anything except fire up a framework that takes care of parsing the*
- rest*
- > *of the URI and passing control over to the proper script. Using some*
- > *relatively cryptic features of Apache you can hide the fact that*
- > *everything goes through index.py (mod_rewrite).*

I have never really liked how all these different mod_python extensions insist on using their own special file extensions, eg., .psp, .mps etc. All it does is to expose what you are using to implement a site and makes it hard for you to convert to a different implementation mechanism as all your URLs then need to change.

As you point out, the only way around it is to use mod_rewrite rules. Overall, from what I have seen all it does is make configuration files harder to understand and spawn lots of newbie questions as to why they can't get anything to work.

To me it is more sensible to use REST principles and not use file type extensions at all, or use an extension which reflects the type of content being delivered up and not the mechanism used to generate it.

FWIW, the Vampire package tries to address this problem in mod_python by providing a simple mechanism as an extension to mod_python which gives one greater control over the file extension used in a URL and what handler that then maps to.

To set it up, one has a single directive in your httpd.conf or htaccess file of the form:

```
PythonHandler vampire
```

comp.lang.python: Re: modpython, apache and windows

You can then place in your directory any number of content handler .py files. Eg. you might have index.py, about.py, report.py etc. In the first instance, no matter what the file type extension on the URL, if the
basename in the URL request matches the basename of the content handler file, then it is that file which is the target of the request.

Thus, if you used the following URLs, they would map as shown:

```
index.html --> index.py
about.html --> about.py
report.pdf --> report.py
report.csv --> report.py
feedback --> feedback.py
```

Now normally the content handler would be called "handler()". Using vampire though, you extend the name with the file type, thus:

```
index.html --> will call handler_html() in index.py
about.html --> will call handler_html() in about.py
report.pdf --> will call handler_pdf() in report.py
report.csv --> will call handler_csv() in report.py
feedback --> will call handler() in feedback.py
```

Thus, you can use a more sensible file type in the URL without having to resort to mod_rewrite rules. Further, where a specific resource can be represented in multiple formats which are both generated on the fly, you can have a handler for each in the same content handler .py file, eg., as in .pdf and .csv case. You can also follow REST principles and have no extension at all.

You aren't restricted to just having content handlers in a directory though. You can still have other files such .html and .jpg files. If Vampire determines that there is no content handler corresponding to a request, it will decline to service it and pass it back to Apache which will then serve up the raw .html or .jpg file directly.

I should point out that Vampire is not a framework in the sense that it doesn't dictate how your pages are rendered. That handler function which is called is basically the same as a stock standard mod_python handler method. Inside that function you still need to provide the code which creates the response, although there is no reason why you can't on a per resource case trigger off a different system such as PSP, mpservlets or HTMLTemplate to generate the HTML.

More could be said, but best to check out:
<http://www.dscpl.com.au/projects/vampire>

--

Re: modpython, apache and windows

comp.lang.python: Re: modpython, apache and windows

Graham Dumpleton