repoze.vhm Documentation

Release 1.5

Agendaless Consulting, Inc.

Apr 06, 2017

Contents:

CHAPTER 1

`repoze.vhm` API

CHAPTER 2

Overview

This package provides middleware and utilities for doing virtual hosting within a WSGI/Repoze environment. It is particularly useful within a `repoze.zope2` environment, where it may be used as an alternative to the classic VirtualHostMonster: method of doing virtual hosting.

CHAPTER 3

Virtual Hosting in a Nutshell

"Virtual hosting" enables dynamic applications to be served from within a larger URL namespace, independent of the physical location of the script files used to serve the application, or the precise layout of objects within the application. In particular, the application and the server collaborate to generate URLs for links in the application, such that the links preserve the "apparent" location of the application.

The simplest case requires no effort at all: links rendered as relative paths from within pages work nicely. However, such links begin to be problematic quickly, e.g. when the page is serving as the default index view for its folder, and the URL does not end in a `/`. In that case, the browser interprets the links relative to the folder's parent, and chaos ensues.

CHAPTER 4

CGI Environment Variables

As used for applications running "inside" Apache (e.g., using `mod_python`), there follwing environment variables are of interest when doing virtual hosting:

`SERVER_NAME`

: the name which the server believes it has.

`HTTP_HOST`

: the apparent hostname of the server (i.e., as passed in the `Host:` header)

`SERVER_PORT`

: the apparent port of the server

`SCRIPT_NAME`

: any path prefix used by Apache to dispatch to the application (as defined via the `ScriptAlias` directive).

`PATH_INFO`

: the remainder of the path, after removing any parts used in dispatch.

CHAPTER 5

repoze.vhm#vhm_xheaders WSGI Filter

When configured as WSGI middleware, this filter will convert the path information in the environment from the "X-Vhm" headers added to the request into the "standard" CGI environment variables outlined above. It will also place repoze.vhm-specific environment variables into the WSGI environment for consumption by repoze.zope2 (or another application which chooses to use its services).

If this filter is placed into the pipeline in front of a Zope 2 application, the standard Virtual Host Monster object (`/virtual_hosting`) may be deleted, as it is no longer necessary. However, it does not need to be deleted; repoze.vhm will work if it is present.

The filter requires no configuration; it can be added to any pipeline via its egg name: `egg:repoze.vhm#vhm_xheaders`.

CHAPTER 6

repoze.vhm#vhm_explicit WSGI Filter

This filter is like the repoze.vhm#vhm_xheaders filter, but instead of taking the virtual host and/or root from the environment, they are explicitly configured when the middleware is instantiated.

If using paste.deploy, this looks like:

```
[filter:vhm]
use = egg:repoze.vhm#vhm_explicit
host = http://www.example.com
root = /mysite
```

Both host and root are optional, but you probably want to specify at least one of them.

CHAPTER 7

repoze.vhm#vhm_path WSGI Filter

As a fallback for proxies which cannot add headers to proxied requests, this filter implements the same path-based virtual hosting syntax used by the Zope2 Virtual Host Monster. Because this syntax is quite arcane (so much that there is a web-app for generating the rewrite rules!), this filter is not recommended except for environments which cannot be configured to add headers (e.g., Apache has `mod_rewrite` enabled, but cannot be changed to enable `mod_headers`).

When configured as WSGI middleware, this filter will convert the path information in the environment from the classic "Zope2 virtual hosting mungned URL" into the "standard" CGI environment variables outlined above. It will also place repoze.vhm-specific environment variables into the WSGI environment for consumption by repoze.zope2 (or another application which chooses to use its services).

If this filter is placed into the pipeline in front of a Zope 2 application, the standard Virtual Host Monster object (`/virtual_hosting`) may be deleted, as it is no longer necessary. However, it does not need to be deleted; repoze.vhm will work if it is present.

The filter requires no configuration; it can be added to any pipeline via its egg name: `egg:repoze.vhm#vhm_path`.

You can set the `conserve_path_infos` parameter if you want only the host and port bits to be touched.:

```
[filter:vhm]
use = egg:repoze.vhm#vhm_path
conserve_path_infos = true
```

This trick can be useful to forward the url AS-IS to an underlying equipment wiwhtout touching to the URI. Eg, a wsgi pipeline where our repoze.vhm filters some content from a stock zope2 server which do the VHM stuff also on its own.

CHAPTER 8

Example: a deliverance middleware filtering out a stock zope2+plone server

We have in this setup a frontal apache server reverse proxyfing a deliverance server.

First, we will tell apache to proxy our requests to our backends.

Apache will tell:

- At the wsgi level, rewrite links using things in the request headers
- At the zope2 level, use the URL with vhmonster to rewrite links.

```
# Rewrite links in paste
RequestHeader add        X-Vhm-Host http://host.tld:80/
# /_themes & .deliverance -> deliverance
ProxyPassMatch   /(_themes|\.deliverance)(.*) http://localhost:8378/$1$2
ProxyPassReverse /_themes/                    http://localhost:8378/_themes/
ProxyPassReverse /.deliverance/               http://localhost:8378/.deliverance/

# application mounted on / does not needs _vh_
# /zmiroot -> access to zmi
ProxyPassMatch   ^/zmiroot(.*) http://localhost:8381/VirtualHostBase/http/host.tld:80/
↪VirtualHostRoot/_vh_zmiroot$1
ProxyPassReverse ^/zmiroot/    http://localhost:8381/VirtualHostBase/http/host.tld:80/
↪VirtualHostRoot/_vh_zmiroot/

# /plone-plonesiteid-> vhmonster without deliverance filtering
ProxyPassMatch   ^/plone-plonesiteid(.*) http://localhost:8381/VirtualHostBase/http/
↪host.tld:80/plonesiteid/VirtualHostRoot/_vh_plone-plonesiteid$1
ProxyPassReverse ^/plone-plonesiteid/    http://localhost:8381/VirtualHostBase/http/
↪host.tld:80/plonesiteid/VirtualHostRoot/_vh_plone-plonesiteid/

# /-> zope2vhmonster
ProxyPass        /               http://localhost:8378/VirtualHostBase/http/host.
↪tld:80/plonesiteid/VirtualHostRoot/
ProxyPassReverse /               http://localhost:8378/VirtualHostBase/http/host.
↪tld:80/plonesiteid/VirtualHostRoot/
```

This deliverance server is just another PasteDeploy setup which query somehow a zope2 server listening somewhere.

```
[DEFAULT]
debug=false

[app:athemes]
use = egg:Paste#static
document_root=%(here)s/sometheme/path

[app:azopeproxy]
use = egg:Paste#proxy
address=http://ZOPE:8381/

[filter:fdeliverance]
use=egg:Deliverance
theme_uri = sometheme
rule_uri = file://somerules
execute_pyref=true

[filter:ftranslogger]
use=egg:Paste#translogger
setup_console_handler=true

[filter:fexc]
use=egg:WebError#evalerror

[filter:fvhm]
use = egg:repoze.vhm#vhm_xheaders

[pipeline:pmain]
pipeline =  fexc
            ftranslogger
            fvhm
            fdeliverance
            azopeproxy

[composite:main]
use = egg:Paste#urlmap
/ = pmain
/_themes = athemes

[server:main]
use=egg:Paste#http
host = localhost
port = 8378
```

CHAPTER 9

repoze.vhm Virtual Hosting Model

This model (based on a suggestion of Ian Bicking's ,), passes virtual hosting information from the proxy / web server to the application by adding extra headers to the proxied request:

`HTTP_X_VHM_HOST`

> indicates the apparent URL prefix of the root of the application (concatenating `wsgi.url_scheme`, `SERVER_NAME`, `SERVER_PORT`, and `SCRIPT_NAME` variables; the equivalent of Zope2's `SERVER_URL`).

`HTTP_X_VHM_ROOT`

> path of the object within the application which is supposed to function as the "virtual root".

When serving an application from "within" Apache via mod_wsgi, we can just set the environment directly:

```
<Directory /path/to/wsgiapp>
  SetEnv HTTP_X_VHM_HOST http://www.example.com
  SetEnv HTTP_X_VHM_ROOT /cms
</Directory>
```

If you are serving repoze.zope2 via a proxy rewrite rule, you may pass this information by adding additional headers. E.g., a sample Apache configuration for the example above might be:

```
<VirtualHost *:80>
  ServerName www.example.com
  RewriteEngine on
  RewriteRule ^/(.*) http://localhost:8080/$1 [P,L]
  RequestHeader add X-Vhm-Host http://www.example.com
  RequestHeader add X-Vhm-Root /cms
</VirtualHost>
```

In either of the above example cases, the effect on repoze.zope2 when repoze.vhm's filter is in the WSGI pipeline is the same: the apparent root of `http://www.example.com` will be the default view of the object that has a physical path of `/cms`. Additionally, paths in URLs generated by Zope will not start with `/cms`, and the scheme and hostname in URLs will be "http://www.example.com" as opposed to `http://localhost:8080`.

The "vhm host" header may contain further path information as necessary; further path information can (and will, in the case of repoze.zope2) be respected by downstream applications to root an application at a non-server-root path:

```
<Directory /path/to/wsgiapp>
  SetEnv HTTP_X_VHM_HOST http://www.example.com/further/path
  SetEnv HTTP_X_VHM_ROOT /cms
</Directory>
```

In this case, URLs generated by Zope will begin with `http://www.example.com/further/path`. This syntax replaces the "inside out" virtual hosting syntax (`_vh_` segment markers in the URL) as described in the "Virtual Host Monster" documentation.

The "vhm host" and "vhm root" headers can be used independently (the system will operate as you would expect in the absence of one or the other).

CHAPTER 10

repoze.vhm Library API

Because the existing Zope 2 virtual hosting machinery does not rely on the "standard" CGI variables, the application dispatcher needs to "fix up" the environment to match Zope's expectations. repoze.vhm offers the following functions to aid in this fixup:

repoze.vhm.utils.setServerURL()

> convert the standard CGI virtual hosting environment into the form expected by Zope2 (adding the SERVER_URL key).

repoze.vhm.utils.getVirtualRoot()

> return the virtual root path (repoze.vhm.virtual_root) as set by the middleware.

CHAPTER 11

Indices and tables

- genindex
- modindex
- search

Index