repoze.errorlog Documentation

Release 1.0

Agendaless Consulting, Inc. and Contributors

June 03, 2016

Contents

1 Using `repoze.errorlog` **3**
- 1.1 Overview . . . 3
- 1.2 Configuration . . . 3
- 1.3 Viewing Recent Tracebacks . . . 4
- 1.4 Integration . . . 4
- 1.5 Reporting Bugs / Development Versions . . . 4

2 `repoze.errorlog` API **5**

3 Indices and tables **7**

Contents:

CHAPTER 1

Using repoze.errorlog

1.1 Overview

This package implements a WSGI middleware filter which intercepts exceptions and writes them to a Python logging module channel (or the `wsgi.errors` filehandle, if no channel is configured). It also allows the browsing of limited exception history via a browser UI.

1.2 Configuration

To use the default configuration, you can just include the filter in your application's PasteDeploy pipeline, e.g.:

```
[pipeline:main]
pipeline = egg:Paste#cgitb
           egg:repoze.errorlog#errorlog
           yourapp
```

To override the default configuration, you need to make a separate section for the filter. The Paste configuration options at this time are `channel`, `keep` and `path`. To configure `repoze.errorlog` to use the `Repoze` logging channel, which sends to the logging channel as if you had send to a logger from code where you did `from logging import getLogger; logger = getLogger("Repoze")` and to keep 50 tracebacks around for through-the-web exception viewing, configure like so:

```
[filter:errorlog]
channel = Repoze
keep = 50
path = /__my_error_log__
ignore = RuntimeError my.module:MyError
```

By default, no channel is configured, and tracebacks are sent to the `wsgi.errors` file handle (which should cause the errors to show up in your server's error log). By default, the exception history length (`keep`) is 20.

By default, the error log's path is `/__error_log__`; you can change this as necessary for your deployment.

The `ignore` parameter prevents the exceptions named from being logged or kept in exception history (although they are reraised). By default, no exceptions are ignored.

To use the reconfigured filter in the pipeline:

```
[pipeline:main]
pipeline = egg:Paste#cgitb
           errorlog
           yourapp
```

If you don't use PasteDeploy, you can configure the ErrorLog middleware manually:

```
app = ErrorLog(app, channel=None, keep=20, path='/__error_log__',
               ignored_exceptions=())
```

1.3 Viewing Recent Tracebacks

To view recent tracebacks via your browser (exception history), visit the `/__error_log__` path at the hostname represented by your server. A view will be presented showing you all recent tracebacks. Clicking on one will bring you to a page which shows you the traceback and a rendering of the WSGI environment which was present at the time the exception occurred.

1.4 Integration

When `repoze.errorlog` is placed into the pipeline, two keys are placed into the wsgi environment on every request (even when an exception is not raised and caught by `repoze.errorlog`):

`repoze.errorlog.path` the path at which the errorlog is configured

`repoze.errorlog.entryid` the entry id of the next error

Middleware and applications that catch exceptions can compose a URL to the current error (for helpful development feedback) when they know `repoze.errorlog` ' is in the pipeline by using the following code:

```
from paste.request import construct_url
path = environ['repoze.errorlog.path']
entry = environ['repoze.errorlog.entryid']
url = construct_url(environ, path_info=path,
                    querystring='entry=%s' % entry)
```

1.5 Reporting Bugs / Development Versions

Please report bugs at https://github.com/repoze/repoze.errorlog/issues . Check development version out from https://github.com/repoze/repoze.errorlog .

CHAPTER 2

repoze.errorlog API

CHAPTER 3

Indices and tables

- genindex
- modindex
- search