dataflake.cache Documentation
Release 1.0dev

Jens Vagelpohl

March 11, 2009

CONTENTS

`dataflake.cache` provides caching implementations based on a very simple API.

NARRATIVE DOCUMENTATION

Narrative documentation explaining how to use `dataflake.cache`.

1.1 Installing `dataflake.cache`

1.1.1 How To Install

You will need [Python](#) version 2.4 or better to run `dataflake.cache`. Development of `dataflake.cache` is done primarily under Python 2.4, so that version is recommended.

> **Warning:** To succesfully install `dataflake.cache`, you will need an environment capable of compiling Python C code. See the documentation about installing, e.g. `gcc` and `python-devel` for your system. You will also need *setuptools* installed on within your Python system in order to run the `easy_install` command.

It is advisable to install `dataflake.cache` into a *virtualenv* in order to obtain isolation from any "system" packages you've got installed in your Python version (and likewise, to prevent `dataflake.cache` from globally installing versions of packages that are not compatible with your system Python).

After you've got the requisite dependencies installed, you may install `dataflake.cache` into your Python environment using the following command:

```
$ easy_install dataflake.cache
```

1.2 Development

1.2.1 Bug tracker

For bug reports, suggestions or questions please use the [Dataflake bug tracker](#).

1.2.2 Source code

The source code is maintained in the Dataflake Subversion repository. To check out the trunk:

```
svn co http://svn.dataflake.org/svn/dataflake.cache/trunk/
```

You can also browse the code online.

When using setuptools or zc.buildout you can use the following URL to retrieve the latest development code as Python egg:

```
http://svn.dataflake.org/svn/dataflake.cache/trunk#egg=dataflake.cache
```

1.3 Glossary

Setuptools Setuptools builds on Python's `distutils` to provide easier building, distribution, and installation of packages.

Interface An attribute of a model object that determines its type. It is an instance of a `zope.interface` Interface class.

Zope The Z Object Publishing Framework. The granddaddy of Python web frameworks.

Virtualenv An isolated Python environment. Allows you to control which packages are used on a particular project by cloning your main Python. virtualenv was created by Ian Bicking.

1.4 `dataflake.cache` Change History

```
Change log
==========

0.9 (unreleased)
----------------

- Initial release based on caching code formerly residing inside
  Products.LDAPUserFolder
```

API DOCUMENTATION

API documentation for `dataflake.cache`.

2.1 `dataflake.cache.interfaces`

Interface: **`dataflake.cache.interfaces.ICache`** Simple cache interface

> Method: **`invalidate(key=None)`** Invalidate the given key, or all key/values if no key is passed.
>
> Method: **`set(key, value)`** Store a key/value pair
>
> Method: **`get(key=None, default=None)`** Get value for the given key, or all values if no key is passed
> If no value is found or the value is invalid, the default value will be returned.

Interface: `dataflake.cache.interfaces.ITimeoutCache`

- Extends: ICache

  Simple cache with a timeout

  Only records younger than the configured timeout are returned

  Method: **`setTimeout(timeout)`** Set a timeout value in seconds

SUPPORT

If you need commercial support for this software package, please contact Zetwork GmbH at http://www.zetwork.com.

INDICES AND TABLES

- *Index*

- *Module Index*

- *Search Page*

- *Glossary*

INDEX

I
Interface, 4

S
Setuptools, 4

V
Virtualenv, 4

Z
Zope, 4