1 PIOUS

Contents

1.1 Introduction

A python package for dealing with basic input/output processes. Pious came from the observation that the processing data almost always boils down to a series of discreet transforms. Splitting these transforms out into reusable modules is not only a useful way to build data processing pipelines but also provide a convenient way to design and describe the process to others.

Pious aims to provide the tools needed to build complex data transformation pipeline in a simple and easily understood manner. It also try to provide the tools necessary to inspect data as it passes through the pipeline to make debugging processes as easy as possible.

2 Pious Data Sources

At the base of all Pious transform pipelines are data sources, which as the name would imply are sources of the raw data you wish to transform.

Pious provides a number of data source primatives along with the provisions to build upon those primatives to fit your own requires.

2.1 Data Source Primitives

Source

The base class for all data sources. This is the class from which you should extend you own custom data sources.

CSVFile

A Character separated values file. You can specify the separator, escape character etc.

XMLFile

The build blocks of a SAX based XML file parser

FlatXMLFile

This extends the XMLFile primitive to provide a simple flat xml importer. By flat XML I mean one that bascially imitates a CSV, it doesn't use attributes, just flat un-nested elements.

DbIterator

Iterators the results of an SQL query

3 Pious Pipes

Pipes are similar to data sources in that they act as iterators. The different is that they take a datasource (or another pipe) as an input. Pipes therefore are the building blocks of your transformation pipeline.

Pious provides a goodly collection of pipe with which to build your pipelines providing filters, field transformers along with more complex pipes like fork which takes a pipeline and splits it into 2 or multiple parallel pipes.

3.1 Existing Pipes

Pipe

The base pipe class that all others extend. This should be the basis of your own pipe implementations

Ensure

Ensure that the dict passing through has certain keys and that if they are not present then we set a default value

Filter

Skips items that match certain criteria

Fork

Splits the pipe into 2 parallel pipes - i.e. the output of the bound iterator is fed into the input of 2 pipelines.

4 Pious Consumer

Consumers are the termination points of pipelines. They consume the final output of the pipeline. For example they may write to a file in various formats or insert/replace into a database table or tables.

4.1 Existing Consumers

Consumer

The base class for all Pious consumers. This is the class from which your own custom consumers should derive.

DbTable

Inserts/Replaces values into a table. Also has options for truncating the table prior to the insert etc.

CsvFile

Writes the data to a CSV file with optional headers and configurable configurable separators and escape characters