ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

EiffelRSS

Project description

Michael Käser <kaeserm@student.ethz.ch>
Martin Luder <luderm@student.ethz.ch>
Thomas Weibel <weibelt@student.ethz.ch>

inf | Informatik
Computer Science

Abstract

EiffelRSS is an Eiffel library to parse RSS. The goal is to provide the Eiffel development community with an easy to use and well structured API for RSS.

The distribution also contains a simple RSS newsfeed reader written with EiffelVision and EiffelRSS.

1 Overview

1.1 Library

EiffelRSS consists of several different clusters.

1.1.1 FETCH

`FETCH` is a class which has features that can fetch data from a source address to a local `STRING` using varoius services. `FETCH` provides a simple interface for the `DATA_RESOURCE` class in EiffelNet.

A valid source address has the following format: `service://address`

Supported services are: file, http, ftp

See figure 1 for an overview of the class.

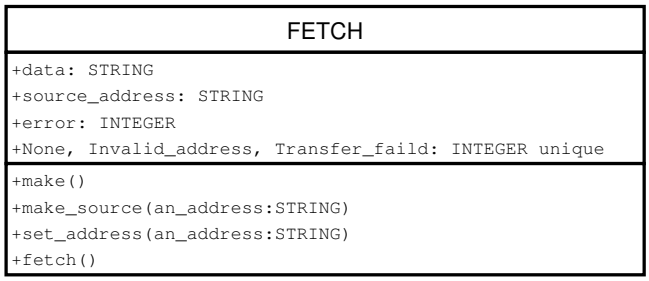

Figure 1: UML diagramm of `FETCH`

1.1.2 LOGFILE

`LOGFILE` represents a file which can be used for logging messages during the program execution.

Each message has its own priority (a positive integer value) and each logfile a certain user defined threshold. If the priority of the message is greater or equal than the threshold, it gets written to the log file together with a timestamp.

See figure 2 for an overview of the class.

1.1.3 PROPERTIES

`PROPERTIES` represents a persistent set of properties. The properties can be saved to a file or loaded from a file.

Each key and its corresponding value in the property list is a string.

A property list can contain another property list as its default. This default property list is searched if the property key is not found in the original property list.

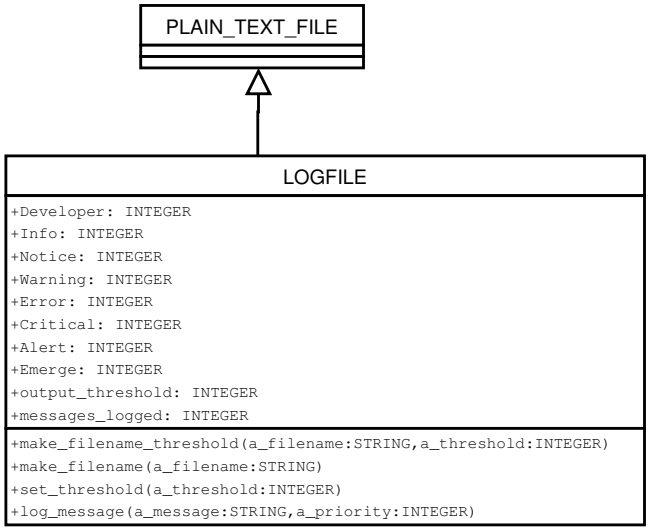

Figure 2: UML diagramm of `LOGFILE`

`PROPERTIES` is similar to the `java.util.Properties` class. The main difference is, that `PROPERTIES` doesn't support keys and values separated by whitespace. It always expects `:` or `=` as a separator between key and value.

See figure 3 for an overview of the class.

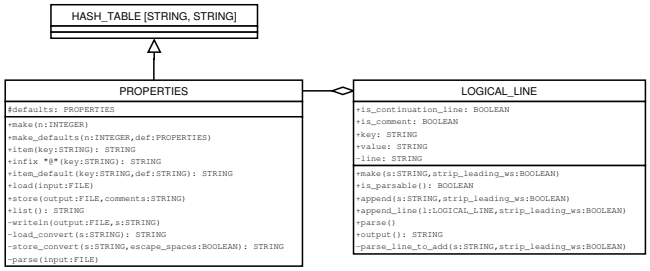

Figure 3: UML diagramm of `PROPERTIES`

1.1.4 SYNDICATION

`SYNDICATION` is the main cluster of EiffelRSS. It contains several subclusters:

- `FEED`
  Is the central datastructure, defines an abstract syndication feed.
- `READERS`
  Defines readers for different syndication formats.
- `WRITERS`
  Defines writers for different syndication formats.

- `UTILS`
  Provides several utilities for parsing and writing, like encoders, decoders, etc.

1.2 Examples

The EiffelRSS distribution contains several examples demonstrating how to use its features. Among them is a simple newsfeed reader written with EiffelVision. See figure 4 for a mockup of the user interface.

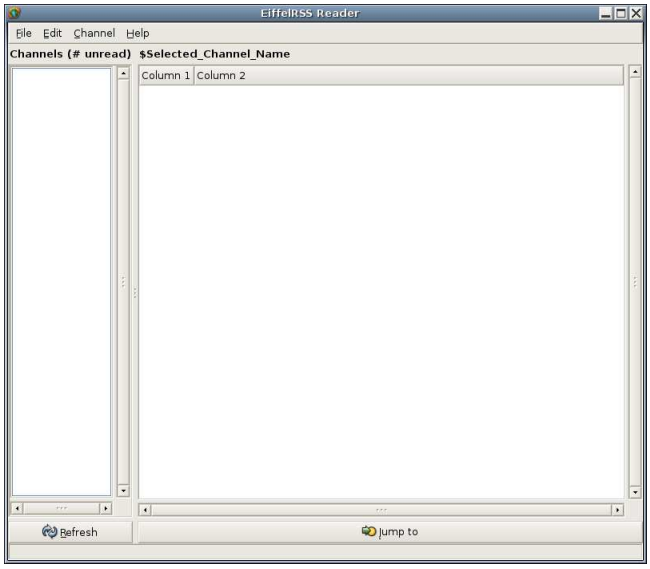

Figure 4: Mockup of the user interface of EiffelRSS Reader

2 Extensibility

EiffelRSS is designed with extensibility in mind.

Initially EiffelRSS will only support reading of RSS 2.0. But the library is designed to be easily extensible by other RSS readers. Even non-RSS syndication formats like Atom can be parsed into EiffelRSS' datamodel. It is also feasible to add writing support and to write any syndication format (even non-XML). Because EiffelRSS uses an abstract intermediate representation of newsfeed data, one can also convert from one format to another.

Another possibility of extension would be the sample newsfeed reader written with EiffelVision. This application could be extended to a full fledged newsfeed reader.

To sum up: EiffelRSS is extensible without limit.