

EiffelRSS

FETCH: Developer Guide

Michael Käser <kaeserm@student.ethz.ch>

Martin Luder <luderm@student.ethz.ch>

Thomas Weibel <weibelt@student.ethz.ch>

Abstract

FETCH is a class which has features that can fetch data from a source address to a local STRING using various services.

FETCH provides a simple interface for the DATA_RESOURCE class in EiffelNet.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Usage	1
2	Features	3
2.1	Initialization	3
2.1.1	make	3
2.1.2	make_source	3
2.2	Access	3
2.2.1	data	3
2.2.2	source_address	3
2.2.3	error	3
2.3	Basic Operations	4
2.3.1	set_address	4
2.3.2	fetch	4

List of Figures

1.1 UML diagram of <code>FETCH</code>	1
---	---

Chapter 1

Introduction

1.1 Overview

FETCH is a class which has features that can fetch data from a source address to a local STRING using various 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 1.1 for an overview of the class.

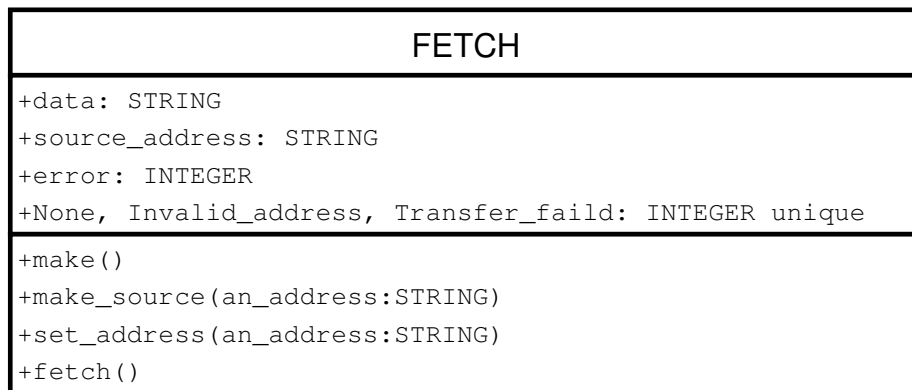


Figure 1.1: UML diagram of FETCH

1.2 Usage

```
class
  USAGE_EXAMPLE
```

```
create
  make

feature — Initialization

  make is
    — Creation procedure.
    local
      fetch: FETCH
      address: STRING
    do
      create fetch.make

      io.put_string ("Please enter an address to fetch: \
- %N")
      io.read_line

      address := io.last_string.twin

      fetch.set_address (address)
      fetch.fetch

      io.put_new_line

      if (fetch.error = fetch.Invalid_address) then
        io.put_string ("Error: Invalid address")
      elseif (fetch.error = fetch.Transfer_failed) then
        io.put_string ("Error: Transfer failed")
      else
        io.put_string (fetch.data)
      end

      io.put_new_line
    end

end — class USAGE_EXAMPLE
```

Chapter 2

Features

2.1 Initialization

2.1.1 make

`make ()`
— *Create the object without a source address*

2.1.2 make_source

`make_source (an_address: STRING) is`
— *Create the object with a predefined source*

2.2 Access

2.2.1 data

`data: STRING`
— *The data fetched*

This can be void if there was an error.

2.2.2 source_address

`source_address: STRING`
— *The source address*

This can be void.

2.2.3 error

`error: INTEGER`
— *An error number*

error can be one of the following constants: `None`, `Invalid_address`, `Transfer_failed`.

`None` means that there was no error and data is available. `Invalid_address` means that the given source address was either empty or not valid. If the error is `Transfer_failed`, there was a problem when trying to load the data, i.e. there was no connection to the internet.

2.3 Basic Operations

2.3.1 `set_address`

```
set_address (an_address: STRING)  
    — Sets the address to an_address
```

This sets the source address to `an_address`. After calling this feature, error can be `Invalid_address`.

2.3.2 `fetch`

```
fetch  
    — Try to fetch the data from source_address
```

If error is not `Invalid_address`, `fetch` tries to open it and load the data. If there is a problem while loading the data, error is set to `Transfer_failed`.