Pipes

From HaskellWiki

The pipes library is a clean and powerful stream processing library that lets you build and connect reusable streaming components.

Contents

1 Libraries

Pipes (http://hackage.haskell.org/packages/archive/pkg-list.html#cat:pipes) is the official Hackage category for the pipes ecosystem.

1.1 pipes

The pipes (http://hackage.haskell.org/package/pipes) package is the core library which provides reusable primitives for stream programming. Use these as building blocks for more sophisticated streaming abstractions.

Read the official pipes tutorial (http://hackage.haskell.org/packages/archive/pipes

/4.0.0/doc/html/Pipes-Tutorial.html) to learn more.

Source code hosted on GitHub (https://github.com/Gabriel439/Haskell-Pipes-Library)

1.2 pipes-safe

The pipes-safe (http://hackage.haskell.org/package/pipes-safe) package adds resource management and exception safety to the pipes ecosystem. Use pipes-safe to safely acquire and release resources deterministically within a pipeline.

Source code hosted on GitHub (https://github.com/Gabriel439/Haskell-Pipes-Safe-Library)

1.3 pipes-concurrency

The pipes-concurrency (http://hackage.haskell.org/package/pipes-concurrency) package adds concurrency primitives to the pipes ecosystem with built-in deadlock safety. Use these primitives to:

- Build reactive event-driven programs
- Merge and broadcast streams
- Communicate between multiple concurrent pipelines

Read the official pipes-concurrency tutorial (http://hackage.haskell.org/packages/archive/pipes-concurrency/2.0.0/doc/html/Pipes-Concurrent-Tutorial.html) to learn more.

Source code hosted on GitHub (https://github.com/Gabriel439/Haskell-Pipes-Concurrency-Library)

1.4 pipes-parse

The pipes-parse (http://hackage.haskell.org/package/pipes-parse) package defines the generic machinery necessary for common parsing tasks. Use pipes-parse to:

- handle leftovers and end of input,
- interrupt and resume streaming, and
- sub-divide streams without collecting elements in memory.

Source code hosted on GitHub (https://github.com/Gabriel439/Haskell-Pipes-Parse-Library)

2 Community-contributed libraries

Listed in alphabetical order.

2.1 pipes-aeson

The pipes-aeson (http://hackage.haskell.org/package/pipes-aeson) library allows you to encode and decode JSON values flowing through streams, possibly interleaving other stream effects while doing it.

Source code is hosted on GitHub (https://github.com/k0001/pipes-aeson)

2.2 pipes-attoparsec

The pipes-attoparsec (http://hackage.haskell.org/package/pipes-attoparsec) library converts attoparsec (http://hackage.haskell.org/package/attoparsec) parsers to pipes for high-performance incremental parsing.

Source code hosted on GitHub (https://github.com/k0001/pipes-attoparsec)

2.3 pipes-binary

The pipes-binary (http://hackage.haskell.org/package/pipes-binary) library allows streams of binary data to be encoded and decoded using the `Binary` instances from the binary (http://hackage.haskell.org/package/binary) package.

Source code is hosted on GitHub (https://github.com/k0001/pipes-binary)

2.4 pipes-network

The pipes-network (http://hackage.haskell.org/package/pipes-network) library converts server and client sockets to pipes to seamlessly stream data over any network.

Source code hosted on GitHub (https://github.com/k0001/pipes-network)

2.5 pipes-network-tls

The pipes-network-tls (http://hackage.haskell.org/package/pipes-network-tls) allows streaming through TLS-secured network connections, exposing a similar API to the one exposed by pipes-network (http://hackage.haskell.org/package/pipes-network) .

Source code hosted on GitHub (https://github.com/k0001/pipes-network-tls)

2.6 pipes-zlib

The pipes-zlib (http://hackage.haskell.org/package/pipes-zlib) enables compression and decompression of strict ByteString streams using the zlib codec.

Source code hosted on GitHub (https://github.com/k0001/pipes-zlib)

3 Community

Besides the usual Haskell community channels such as the official mailing lists or the Haskell subreddit (http://reddit.com/r/haskell) , you can ask for help, suggest improvements, or discuss about the Pipes ecosystem at the “haskell-pipes” mailing list at Google Groups (https://groups.google.com/group/haskell-pipes) .

4 Announcements

In chronological order:

- pipes-1.0.0 (http://www.reddit.com/r/haskell/comments/ohjg7/a_new_approach_to_iteratees/) : The original announcement, which first introduced the concept of unifying sources and sinks and transducers into a single Category (http://www.haskell.org/ghc/docs/latest/html/libraries/base/Control-Category.html) .

- pipes-2.0.0 (http://www.haskellforall.com/2012/05/pipes-20-pipe-finalization.html) : The introduction of Frames, later deprecated in favor of pipes-safe.

- pipes-2.1.0 (http://www.haskellforall.com/2012/07/pipes-21-and-index-core-10-indexed.html) : Transition of Frames to indexed monads, later deprecated in favor of pipes-safe.

- pipes-2.3.0 (http://www.haskellforall.com/2012/09/pipes-23-bidirectional-pipes.html) : The introduction of bidirectional Proxies, which evolved into modern pipes.

- pipes-2.4.0 (http://www.haskellforall.com/2012/10/pipes-24-proxy-transformers-extra.html) : The release of the proxy transformer extension system.

- pipes-2.5.0 (http://www.haskellforall.com/2012/10/pipes-25-faster-and-slimmer.html) : Major performance improvements.

- pipes-3.0.0 (http://www.haskellforall.com/2012/12/pipes-30-simpler-unified-api.html) : Major API simplification and consolidation.

- pipes-safe-1.0.0 (http://www.haskellforall.com/2013/01/pipes-safe-10-resource-management-and.html) : Resource safety and exception management

- pipes-network-0.1.0 (http://www.haskell.org/pipermail/haskell-cafe/2013-March/106752.html) : Use TCP network sockets as pipes streams.

- pipes-3.2.0 (http://www.haskellforall.com/2013/03/pipes-32-listt-codensity-arrowchoice.html) : ListT integration, Codensity proxy transformer, and ArrowChoice primitives.

- pipes-concurrency-1.0.0 (http://www.haskellforall.com/2013/04/pipes-concurrency-100-reactive.html) : Concurrency support for pipes

- pipes-zlib-0.2.0.0 (https://groups.google.com/forum/?fromgroups=#!topic/haskell-pipes/O4_5dZ7WwXE) : Zlib compression/decompression support for pipes

- pipes-network-tls-0.1.0.0 (https://groups.google.com/d/msg/haskell-pipes/XGb-8x5YWY0/41BUbJagnuEJ) : Stream through TLS-secured network connections.

- pipes-parse-1.0.0 (http://www.haskellforall.com/2013/06/pipes-parse-100-pushback-delimited.html) : Pushback, delimited parsers, resumable parsing, and lenses

- pipes-attoparsec-0.2.0.0 (https://groups.google.com/d/msg/haskell-pipes/3A-RbzCUovw/9GEIDi5WRbwJ) : New API, interleaved parsing support, built on top of pipes-parse.

- pipes-aeson-0.1.0.0 (https://groups.google.com/d/msg/haskell-pipes/2CQ9eiPUxjA/kBqcQboPdjoJ) : Encode and decode JSON streams.

- pipes-binary-0.1.0.0 (https://groups.google.com/d/msg/haskell-pipes/7I8KcMW3zHU/5Mi8PKbuv2sJ) : Encode and decode binary streams using the `binary` package.

- pipes-4.0.0 (http://www.haskellforall.com/2013/09/pipes-40-simpler-types-and-api.html) : Simpler types and API

5 Upcoming libraries

- pipes-bytestring: ByteString support. Source code is hosted at GitHub (https://github.com/Gabriel439/Haskell-Pipes-ByteString-Library)

- pipes-text: Text support

6 Videos

- Pipes (http://www.youtube.com/watch?v=2jdJGdA7AYs) , by Oliver Charles. London Haskell user group, 18th September 2013.

7 See also

- The core flaw of pipes and conduit (http://www.yesodweb.com/blog/2013/10/core-flaw-pipes-conduit) (blog article)

Retrieved from "https://wiki.haskell.org/index.php?title=Pipes&oldid=57024"
Categories:

- Idioms
- Libraries
- Packages
