Pipeline-D™ v1.0 Operations Manual

About

Pipeline-D™ (powered by ProtoSlicer® and BitAtomizer®) is a cloud-based encoding and decoding solution designed to reduce data bandwidth, transmission rates and processing costs more optimally than standard industry solutions, without loss of information. It is designed to be set up on each side of a communication channel, without any code refactoring.

To achieve its level of performance, Pipeline-D™ combines several optimization techniques, including:

- Automatically converting data in real time and at a variable bitrate using our patent-pending technologies ProtoSlicer® and BitAtomizer®
- Preserving active sessions in live memory to prevent redundant data of being repeated between messages
- No deserialization or data validation during processing
- Being compatible with traditional compression algorithms for optimal savings

Pipeline-D™ natively supports JSON, with additional data formats planned.

Setup

Pipeline-D™ should be set up on at least 2 Linux-based container environments running in different locations, specifically the sender and the recipient. An encoding container should be set up on the sender side, while a decoding container should be set up on the recipient side.

To run multiple encoding and/or decoding sessions in parallel, make sure to deploy at least one container pair per potential parallel session.

IMPORTANT: Pipeline-D™ is designed to run in a closed cloud ecosystem isolated from third parties, and to execute only one request at a time. In addition, Pipeline-D™ assumes the input data it receives to be valid to maximize performance. If these assumptions are violated, it should be considered a security vulnerability which may cause data leakage, data corruption, denial of service and/or remote code execution.

General usage

Pipeline-D™ containers automatically listen to HTTP requests on port 8080 when run.

Pipeline-D™ is designed to continuously be in a single stateful session. A session is defined as a linear series of messages to be shared between the same sender and recipient. Pipeline-D™ preserves the current session in memory over separate requests until it is reset through a

dedicated endpoint. As each session builds a dictionary of processed values in volatile memory, longer sessions generally yield better reduction results. In the case where the dictionary becomes too large, a session reset will free up the related memory when needed.

Whenever a compatible data format is generated on the sender's side, instead of sending it directly to the recipient, the following procedure should be applied:

1. The sender requests its encoder container to reset its state, if applicable
2. The sender forwards the message to its encoder container
3. The sender forwards the encoded message to the recipient
4. The recipient requests its decoder container to reset its state if the sender did so in step 1
5. The recipient forwards the encoded message to its decoder container
6. The recipient forwards the decoded message to its intended destination
7. Steps 2 to 5 are to be repeated until all messages in the session are processed

Refer to the API definition below for details on how to execute each step.

API reference

POST /compress

Reduces and encodes a message with ProtoSlicer® and BitAtomizer®.

Input:

- Content-Type: `application/json`
- Body: Arbitrary JSON text as defined in RFC 8259 and/or ECMA-404

Output:

- Content-Type: `application/octet-stream`
- Body: The encoded data, based on data already reduced within the current session

Note that the input JSON text is automatically minified as part of the reduction process to maximize performance, which may result in additional savings for the recipient later when they decode this endpoint's output back to JSON text.

IMPORTANT: If the request body contains invalid data, the behavior of this endpoint is undefined until the next session reset. In particular, ensure that:

- **All strings are escaped correctly.**
- **There are no trailing commas at the end of JSON objects or arrays.**
- **Non-numeric floating-point values such as NaN and Infinity are serialized to valid JSON values. (For example, Python's standard library generates invalid JSON by default in this case.)**

POST /decompress

Decodes data reduced with ProtoSlicer® and BitAtomizer® back to its original format.

Input:

- Content-Type: `application/octet-stream`
- Body: A message previously encoded with Pipeline-D™ to be restored

Output:

- Content-Type: `application/json`
- Body: The restored data

IMPORTANT: Requests to decode messages must be done in the same order as they were encoded, and messages encoded from different sessions cannot be decoded in the same session, otherwise the behavior of this endpoint is undefined until the next session reset.

GET /reset

Alias of POST /reset

POST /reset

Resets the encoding and/or decoding session, freeing all memory associated with it.

Input: N/A

Output: HTTP 200 OK on success.

Note that this endpoint can also be used to test if the service is ready.

Changelog

- v1.0: Initial release.