

Node.js Storage.Timeline Dump library

Document Version: 1.1.

Author: Andrew.

Date: 2023.11.10.



This library provides classes for dumping and restoring Storage.Timeline data. It allows exporting and importing Storage.Timeline data as ZIP archives via files or HTTP.

Classes

StreamStorage

Base class for dumping and restoring Storage.Timeline data. Provides methods for streaming data to and from a ZIP archive.

Methods

- `toStream(output)` : Writes a ZIP archive of the storage folder to the given output stream.
- `empty()` : Recursively deletes all files and folders in the storage path.
- `fromStream(input)` : Restores storage folder data from a ZIP archive input stream. (Not implemented in base class).

FileStreamStorage

Extends StreamStorage to read/write Storage.Timeline data to a ZIP file on disk.

Additional Methods

- `toFile()` : Creates a ZIP archive file containing the storage folder data.
- `fromFile(archivePath)` : Restores storage folder data from a ZIP file.
- `removeFile()` : Deletes the ZIP archive file.

HTTPStreamStorage

Extends StreamStorage to read/write Storage.Timeline data to/from a remote HTTP ZIP stream.

Additional Methods

- `fromURI(uri)` : Downloads and restores a ZIP archive from a remote HTTP URI.
- `deploy()` : Restores storage data from a temporary folder to final location.
- `deployTemporary()` : Restores storage data to a temporary folder.

Usage

```
const dump = require("storage-timeline-dump");

// Export storage to ZIP file
const fileStorage = new dump.FileStreamStorage("./storage");
await fileStorage.toFile();
```

```
// Import storage from ZIP file
await fileStorage.fromFile("./storage.zip");

// Import storage from HTTP ZIP stream
const httpStorage = new dump.HTTPStreamStorage("./storage");
await httpStorage.fromURI("https://example.com/storage.zip");
```

Tests

This library includes sample test cases under `tests/` for dumping storages to files and HTTP, and merging storages.