Abstract

This document describes a standardized container format for storing serializable metadata. It does not describe any additional new format, but provides a shell for the exchange of arbitrary, structured data. It shall provide the possibility to store and share any kind of metadata, including encryption support. The idea is to create an open, universal and interoperable standard for storing and distributing every kind of metadata independent from media type or file format.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8335.

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1 Introduction

Nowadays a variety of media files are shared and published all over the world. Information about the origin, purpose or copyright of these files is getting more and more important. There are already different standards which enhance files with metadata, like ID3 [http://id3.org/], Exif [https://www.exif.org/] or Dublin Core [RFC5013]. SimpleMetadata shall create the foundation to unite these standards and provide an universal and open container format. The idea is not to describe any additional new format, but provides a shell for the exchange of arbitrary, structured data. Any metadata payload shall be created, stored and shared with an open standard, like JavaScript Object Notation (JSON) [RFC7159] in combination with schema validation [https://tools.ietf.org/id/draft-handrews-json-schema-00.txt]. Every data structure of metadata can be defined and distributed within schema definitions. Furthermore, SimpleMetadata can be extended by additional formatters or crypto standards.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
2. Structure

The SimpleMetadata format consists of the three parts, Header, Schema URI and Metadata payload.

+-------------------------------------------------------------------+
|            Header (12 bytes) (fixed length)                      |         |
| +-------------------------------------------------------------------+
|            Schema URI (variable length, optional)                |
| +-------------------------------------------------------------------+
|            Serialized Metadata (variable length)                  |
| +-------------------------------------------------------------------+

2.1 Header Definition

+-------------------------------------------------------------------+
| Identifier "SMD" 3 Bytes / string                                  |
| +-------------------------------------------------------------------+
| Version 1 Byte / number                                           |
| +-------------------------------------------------------------------+
| Serialization 1 Byte / number                                     |
| +-------------------------------------------------------------------+
| Crypto 1 Byte / number                                            |
| +-------------------------------------------------------------------+
| Schema URI Length 2 Bytes / number                                |
| +-------------------------------------------------------------------+
| Content Length 4 Bytes / number                                   |
| +-------------------------------------------------------------------+

2.2 Identifier

The first three bytes of the header are always "SMD" to check if SimpleMetadata is present.

2.3 Version

Defines the used version of the SimpleMetadata format.
1 = Current version

2.4 Serialization

Defines the used formatter for the metadata. A formatter serializes or deserializes the metadata with the corresponding procedure. The standard formatter uses Binary JSON (BSON) <http://bsonspec.org/> with the JavaScript Object Notation (JSON) [RFC7159].

0 = BSON

2.5 Crypto
Defines the used crypto standard for en/decrypting metadata.

0 = None encryption
1 = Advanced Encryption Standard (AES)

2.6 Schema URI length

Defines the string length of a schema or type. If no schema is defined the schema length is 0. The schema information is described in chapter 3.

2.6 Content Length

Defines the length of the serialized metadata, based on the selected formatter (Chapter 2.3). If the content is encrypted, the length is calculated over the encrypted string.

3 Schema information

The schema information can be used to validate the metadata against a schema or type. A schema information is an optional string with variable length encoded with UTF-8. It is recommended to use an URI, e.g. "http://exampleschemas.org/Person". Moreover a local file path or even a type definition (AssemblyQualifiedName) can be used. For interoperability, metadata should be based on public schema.

The length of the schema is stored in the header (See chapter 2.6).

4 Metadata Content

Basically every serializable content can be stored as metadata. It is highly recommended to use for interoperability and compatibility the JavaScript Object Notation (JSON) [RFC7159] for metadata and the according schema definition.

5 Notes

Adding SimpleMetadata to a file will damage it under circumstances, unless there is a suitable parser to handle the format!
6 Security Considerations

Sensitive metadata can be encrypted within a supported crypto standard (Chapter 2.4).

7 IANA Considerations

All data must be stored in UTF-8 [RFC2044].

8 References

8.1 Informative References

[RFC7159] Bray, Tim
The JavaScript Object Notation (JSON) Data Interchange Format, March 2014,

The Dublin Core Metadata Element Set, August 2007

[RFC2044] Francois, Yergeau
UTF-8, a transformation format of Unicode and ISO 10646, October 1996

[Internet-Draft] Wright, A.
JSON Schema: A Media Type for Describing JSON Documents, November 19, 2017

Authors’ Addresses

Ralf Dorfner
Muensterplatz 8
88250 Weingarten
Germany

EMail: smd@simplemetadata.org