Network Working Group Internet-Draft Intended Status: Informational Expires: January 5, 2015

The text/markdown Media Type draft-seantek-text-markdown-00.txt

Abstract

This document registers the text/markdown media type for use with Markdown, a family of plain text formatting syntaxes that optionally can be converted to formal markup languages such as HTML.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <u>http://datatracker.ietf.org/drafts/current/</u>.

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."

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <u>BCP 78</u> and the IETF Trust's Legal Provisions Relating to IETF Documents (<u>http://trustee.ietf.org/license-info</u>) 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

In computer systems, textual data is stored and processed using a continuum of techniques. On the one end is plain text: a linear sequence of characters in some character set (code), possibly interrupted by line breaks, page breaks, or other control characters. Plain text provides /some/ fixed facilities for formatting instructions, namely codes in the character set that have meanings other than "represent this character on the output medium"; however, these facilities are not particularly extensible. Compare with [RFC6838] Section 4.2.1. (Applications may neuter the effects of these special characters by prohibiting them or by ignoring their dictated meanings, as is the case with how modern applications treat most control characters in US-ASCII.) On this end, any text reader or editor that interprets the character set can be used to see or manipulate the text. If some characters are corrupted, the corruption is unlikely to affect the ability of a computer system to process the text (even if the human meaning is changed).

On the other end is binary format: a sequence of instructions intended for some computer application to interpret and act upon. Binary formats are flexible in that they can store non-textual data efficiently (perhaps storing no text at all, or only storing certain kinds of text for very specialized purposes). Binary formats require an application to be coded specifically to handle the format; no partial interoperability is possible. Furthermore, if even one byte or bit are corrupted in a binary format, it may prevent an application from processing any of the data correctly.

Between these two extremes lies formatted text, i.e., text that includes non-textual information coded in a particular way, that affects the interpretation of the text by computer programs. Formatted text is distinct from plain text and binary format in that the non-textual information is encoded into textual characters, which are assigned specialized meanings /not/ defined by the character set. With a regular text editor and a standard keyboard (or other standard input mechanism), a user can enter these textual characters to express the non-textual meanings. For example, a character like "<" no longer means "LESS-THAN SIGN"; it means the start of a tag or element that affects the document in some way.

On the formal end of the spectrum is markup, a family of languages for annotating a document in such a way that the annotations are syntactically distinguishable from the text. Markup languages are (reasonably) well-specified and tend to follow (mostly) standardized syntax rules. Examples of markup languages include SGML, HTML, XML, and LaTeX. Standardized rules lead to interoperability between markup processors, but a skill requirement for new (human) users of the

language that they learn these rules in order to do useful work. This imposition makes markup less accessible for non-technical users (i.e., users who are unwilling or unable to invest in the requisite skill development).

informal /-----\ formal plain text informal markup formal markup binary format (Markdown) (HTML, XML, etc.)

Figure 1: Degrees of Formality in Data Storage Formats for Text

On the informal end of the spectrum are lightweight markup languages. In comparison with formal markup like XML, lightweight markup uses simple syntax, and is designed to be easy for humans to enter with basic text editors. Markdown, the subject of this document, is an /informal/ plain text formatting syntax that is intentionally targeted at non-technical users (i.e., users upon whom little to no skill development is imposed) using unspecialized tools (i.e., text boxes). Jeff Atwood once described these informal markup languages as /humane/.[HUMANE]

Markdown specifically is a family of syntaxes that are based on the original work of John Gruber with substantial contributions from Aaron Swartz, released in 2004. [MARKDOWN] Since its release a number of web or web-facing applications have incorporated Markdown into their text entry systems, frequently with proprietary extensions. Fed up with the complexity and security pitfalls of formal markup languages (e.g., HTML5) and proprietary binary formats (e.g., commercial word processing software), yet unwilling to be confined to the restrictions of plain text, many users have turned to Markdown for document processing. Whole toolchains now exist to support Markdown for online and offline projects.

Due to Markdown's intentional informality, there is no standard specifying the Markdown syntax, and no governing body that guides or impedes its development. Markdown works for users for two key reasons. First, the markup instructions (in text) look similar to the markup that they represent; therefore the cognitive burden to learn the syntax is very low. Second, the primary arbiter of the syntax's success is *running code*. The tool that converts the Markdown to a presentable format, and not a series of formal pronouncements by a standards body, is the basis for whether syntactic elements matter.

To support identifying and conveying Markdown (as distinguished from plain text), this document defines a media type and a "flavor" parameter that indicates, in broad strokes, the author's intent on how to interpret the Markdown.

[Page 3]

1.1. Requirements Terminology

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

2. Markdown Media Type Registration Applications

This section provides the media type registration application for the text/markdown media type (see [RFC6838], Section 5.6).

Type name: text

Subtype name: markdown

Required parameters: charset. Per Section 4.2.1 of [RFC6838], charset is REQUIRED. The default value is UTF-8. If omitted, parsers MAY reject the input; if parsers accept the input, they MUST interpret the content as UTF-8.

Optional parameters:

flavor=f; where f is an identifier that specifies the "flavor", or variation, of the Markdown syntax. The parameter represents the intent of the author, namely, that the Markdown will be interpreted "best" (i.e., as the author intended) when processed with tools associated with the identified flavor.

The flavor parameter is opaque and case-sensitive. Valid flavor values can be any sequence of characters or bytes; in practice, however, virtually all will be alphanumeric (US-ASCII) and registered in the IANA Markdown Flavors Registry, discussed in Section 4. Implementations checking flavor parameters MUST only compare them for exact equality.

Encoding considerations: Text.

Security considerations:

Markdown interpreted as plain text is relatively harmless. A text editor need only display the text. The editor SHOULD take care to handle control characters appropriately, and to limit the effect of the Markdown to the text editing area itself; malicious Unicodebased Markdown could, for example, surreptitiously change the directionality of the text. An editor for normal text would already take these control characters into consideration, however.

Markdown interpreted as a precursor to other formats, such as HTML,

[Page 4]

carry all of the security considerations as the target formats. For example, HTML can contain instructions to execute scripts, redirect the user to other webpages, download remote content, and upload personally identifiable information. Markdown also can contain islands of formal markup, such as HTML. These islands of formal markup may be passed as-is, transformed, or ignored (perhaps because the islands are conditional or incompatible) when the Markdown is interpreted into the target format. Since Markdown may have different interpretations depending on the tool and the environment, a better approach is to analyze (and sanitize or block) the output markup, rather than attempting to analyze the Markdown.

Interoperability considerations:

Markdown flavors are designed to be broadly compatible with humans ("humane"), but not necessarily with each other. Therefore, syntax in one Markdown flavor may be ignored or treated differently in another flavor. The overall effect is a general degradation of the output, proportional to the quantity of flavor-specific Markdown used in the text. When it is desirable to reflect the author's intent in the output, stick with the flavor identified in the flavor parameter.

Published specification: This specification.

Applications that use this media type:

Markdown conversion tools, Markdown WYSIWYG editors, and plain text editors and viewers; target markup processors indirectly use Markdown (e.g., web browsers for Markdown converted to HTML).

Additional information:

Magic number(s): None File extension(s): .md, .markdown Macintosh File Type Code(s): TEXT

Person & email address to contact for further information:

Sean Leonard <dev+ietf@seantek.com>

Restrictions on usage: None.

Author: Sean Leonard <dev+ietf@seantek.com>

Intended usage: COMMON

Change controller: The IESG <iesg@ietf.org>

3. Example

The following is an example of Markdown as an e-mail attachment:

MIME-Version: 1.0 Content-Type: text/markdown; charset=UTF-8; flavor=GitHub Content-Disposition: attachment; filename=readme.md

Sample GitHub Markdown

This is some sample GitHub Flavored Markdown (*GFM*). The generated HTML is then run through filters in the [html-pipeline](<u>https://github.com/jch/html-pipeline</u>) to perform things like [sanitization](#html-sanitization) and [syntax highlighting](#syntax-highlighting).

```
Bulleted Lists
```

Here are some bulleted lists...

- * One Potato
- * Two Potato
- * Three Potato
- One Tomato
- Two Tomato
- Three Tomato

More Information

[.markdown, .md](<u>http://daringfireball.net/projects/markdown/</u>) has more information.

<u>4</u>. IANA Considerations

IANA is asked to register the media type text/markdown in the Standards tree using the application provided in <u>Section 2</u> of this document.

Exp. January 5, 2015 [Page 6]

IANA is also asked to establish a subtype registry called "Markdown Flavors". Entries in these registries is by Expert Review [RFC5226]. The Expert will determine whether the registration represents a bonafide variation of the Markdown syntax (i.e., neither a duplicate of an existing registration nor a syntax that is something other than Markdown; [MARKDOWN] SHALL be treated as a normative basis), a brief description, one or more responsible parties, whether the flavor is being maintained at the time of registration, and the existence of at least one complete tool (with or without documentation) that processes the Markdown syntax into a formal document language.

A responsible party can be an individual author or maintainer, a corporate author or maintainer (plus an individual contact), or a representative of a community of interest dedicated to the Markdown syntax.

The registry shall have one initial value, "Standard", with the following data:

Description:

The Markdown syntax as it exists in the Markdown 1.0.1 Perl script at <<u>http://daringfireball.net/projects/markdown/</u>>, with accompanying documentation at <<u>http://daringfireball.net/projects/markdown/syntax</u>>.

Responsible Parties:

(individual) John Gruber <http://daringfireball.net/> <comments@daringfireball.net>

Currently Maintained? No

Tool: Name: Markdown 1.0.1 Reference: <http://daringfireball.net/projects/markdown/> Purpose: Converts to HTML or XHTML circa 2004.

5. Security Considerations

See the answer to the Security Considerations template questions in Section 2.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", <u>BCP 14</u>, <u>RFC 2119</u>, March 1997.

[Page 7]

- [RFC5226] Narten, T., and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", <u>RFC 5226</u>, May 2008.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", <u>BCP 13</u>, <u>RFC</u> <u>6838</u>, January 2013.

6.2. Informative References

- [HUMANE] Atwood, J., "Is HTML a Humane Markup Language?", WWW http://blog.codinghorror.com/is-html-a-humane-markuplanguage/, May 2008.
- [MARKDOWN] Gruber, J., "Daring Fireball: Markdown", WWW <u>http://daringfireball.net/projects/markdown/</u>, December 2004.

Exp. January 5, 2015 [Page 8]

Author's Address

Sean Leonard Penango, Inc. 5900 Wilshire Boulevard 21st Floor Los Angeles, CA 90036 USA

EMail: dev+ietf@seantek.com URI: <u>http://www.penango.com/</u>