HTTP User Agent Feature Identifiers
draft-mcsherry-hafi-00

Abstract

HTTP User Agent Feature Identifiers (HAFI, Hah-Fee) is a simple method for user agents, such as web browsers, to reveal the technologies that are supported by them. These technologies include items such as Cascading Style Sheets, and various versions of HTML.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

HTTP User Agent Feature Identifiers (HAFI, Hah-Fee) is a simple method for user agents, such as web browsers, to reveal the technologies that are supported by them. These technologies include items such as Cascading Style Sheets, and various versions of HTML.

HAFI allows web developers to easily determine the technologies supported by user agents, and present content appropriate to those technologies. HAFI is intended to prevent having to maintain a database of User Agent strings in order to determine functionality.

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 **[RFC2119]**.

2. Implementation

Identifiers are composed of a 32-bit unsigned integer, each bit in this representing a technology. Identifiers are in a little-endian format, and setting a bit to high (one) shows that the technology is supported, while setting a bit to low (zero) shows that the technology is unsupported.

2.1. For User Agents

Identifiers are transmitted to the server via HTTP request headers, using a header entitled "HAFI". This header will provide both the version of HAFI being adhered to, and the identifier itself.

The format of the HAFI header is as shown below. The order of the attributes, "version" and "features", is unimportant, but user agents are required to include both attributes.

HAFI: version=1.0; features=82

User agents should not misrepresent the technologies supported by them, and they should not add vendor-specific values to feature identifiers, as this would render HAFI pointless.

User agents are also advised to set the value of reserved bits to low.

2.2. For Servers and Responders

When implementing HAFI into a server or responder, developers must include every version of HAFI to ensure backwards compatibility with older software that may not support the latest versions of HAFI.

Servers and responders must not attempt to parse any bits that are listed as reserved in the version of HAFI presented by the user agent. While user agents are recommended not to set any reserved bits to high, there is no guarantee that all implementing user agents will do this.

If the presented version of HAFI is a version that is newer than the most recent version implemented in a server or responder, servers and responders should attempt to parse identifiers with the most recent version available.

If the presented version is invalid (e.g. "version=1-t!"), servers and responders should not attempt to parse the features, and are recommended to respond with the HTTP 400 status code (Bad Request).

3. Features

Below is a table listing the features that can be represented in the current version of HAFI. Developers are reminded that features should be presented in a little-endian format, and that reserved bits should be set to low (zero).

Bit(s)	Technology / Description
0	Hypertext Markup Language 4.01
1	Hypertext Matkup Language 5
2	Cascading Style Sheets
3	Cascading Style Sheets 2
4	Cascading Style Sheets 3
5	Cascading Style Sheets 4
6	ECMAScript / JavaScript
7	HTML5 Geolocation
8	HTML5 Video - Ogg Vorbis
9	HTML5 Video - H.264
10	HTML5 Video - WebM
11	HTML5 Canvas
12	Secure Sockets Layer v2
13	Secure Sockets Layer v3
14	Transport Layer Security 1.0
15	Transport Layer Security 1.1
16	Transport Layer Security 1.2
17	HTTP Strict Transport Security
18	Do Not Track (DNT)
19:31	Reserved for Future Usage

It should be noted that, for identifiers such as Do Not Track, the value of that bit being set to high (one) does not indicate that the feature is enabled, only that the user agent supports it.

3.1. HTTP HAFI Header Example

This section will provide an example of a HAFI header that can be used for reference.

The example user agent supports the following technologies: HTML4.01, HTML5, CSS1+2+3, JavaScript, Geolocation, H.264 and WebM, SSLv3, and TLS 1.0 + 1.1. The features identifier in binary would be the following:

00000000 00000000 11100110 11011111

The example user agent is abiding by HAFI 1.0, and the identifier in binary will be converted to decimal for transmission, so the HAFI header sent by the user agent would be the following:

HAFI: version=1.0; features=59103

4. Security Considerations

HAFI should not have any security implications, as it is simply a number and a version string being sent to a server. Rather, HAFI may improve security, in certain contexts.

For example, if a client shows that it supports SSLv3 and TLS 1.0 whilst connecting over HTTPS, the server would be able to select a stream cipher over a block cipher, which would lower the likelihood of a BEAST attack on the client.

5. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).

Author's Address

Liam James McSherry
Email: liamj.mcsherry@googlemail.com