Network Working Group
Internet-Draft
Intended status: Informational
Expires: 20 October 2026

G. Illyes
Google LLC.

18 April 2026

Robots Exclusion Protocol Extension for URL Level Control
draft-illyes-repext-03

Abstract

This document extends RFC9309 by specifying additional URL level controls through an HTTP response header and, for historical reasons, through HTML meta tags originally developed in 1996. Additionally it moves the HTTP response header out of the experimental header space (i.e., "X-") and defines the combinability of multiple headers, which was previously not possible.

Status of This Memo

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on 20 October 2026.

Table of Contents

1. Introduction

While the Robots Exclusion Protocol [ROBOTSTXT] enables service owners to control how, if at all, automated clients known as crawlers may access the URLs on their services as defined by [WEBLINKING], the protocol doesn’t provide controls on how the data returned by their service may be used upon allowed access.

Originally developed in 1996 and widely adopted since, the use-case control is left to URL level controls implemented in the response headers, or in case of HTML in the form of a meta tag as defined by [HTML-META]. This document specifies these control tags, and in case of the response header field, brings it to standards compliance with [HTTP-SEMANTICS].

Application developers are requested to honor these tags. The tags are not a form of access authorization however.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This specification uses the following terms from [STRUCTURED-FIELD-VALUES]: List, String, Parameter.

3. Specification

3.1. Robots control

The URL level crawler controls are a key-value pair that can be specified two ways:

* an HTTP response header structured field as specified by [STRUCTURED-FIELD-VALUES].

* for historical reasons, in case of HTML, one or more meta tags as defined by the [HTML-META] specification.

3.1.1. HTTP Response Header

The robots-tag field is a List as defined in [STRUCTURED-FIELD-VALUES]. Each member of the List is an Item representing a product token. Rules applicable to a product token are defined as Parameters of that Item. For historical reasons, implementors SHOULD also support the experimental field name X-Robots-Tag.

The product token is either a specific string as defined in Section 2.2.1 of [ROBOTSTXT] or the global identifier *. All rules defined in this specification are restrictive. The absence of a rule for a specific instruction constitutes as no instruction.

If a product token appears multiple times in the field, or if rules are provided for both a specific product token and the global * identifier, the implementor MUST apply the union of all restrictive rules. A restriction applied to the global * token applies to all accessors regardless of whether a specific product token is present.

For example, the following response header field specifies a global nosnippet rule for all accessors, and an additional noindex rule specifically for ExampleBot:

Robots-Tag: *;nosnippet, ExampleBot;noindex

The structured field in the examples is deserialized into the following objects:

```
"*" = [
        ["nosnippet", true]
      ],
"ExampleBot" = [
        ["noindex", true],
        ["nosnippet", true]
      ]
```

Implementors SHOULD impose a parsing limit on the field value to protect their systems. The parsing limit MUST be at least 8 kibibytes [KiB].

For a robots-tag field that exceeds the implementor's parsing limit, the implementor MUST process the data up to that limit. Any complete and valid List members found within the processed bytes MUST be honored. Any partially transmitted or truncated List member at the limit, and all subsequent bytes in that field, MUST be ignored. This ensures consistency with the processing of robots.txt files as specified in Section 2.1.1 of [ROBOTSTXT].

3.1.2. HTML Meta Element

For historical reasons the robots-tag header values may be specified by HTML service owners as an HTML meta tag. In case of the meta tag, the name attribute is used to specify the product token, and the content attribute to specify the comma separated robots-tag rules.

As with the header, the product token may be a global token, robots, which signifies that the rules apply to all requestors, or a specific product token applicable to a single requestor. For example:

```
<meta name="robots" content="noindex">
<meta name="examplebot" content="nosnippet">
```

Multiple robots meta elements may appear in a single HTML document. The implementor MUST apply the union of all rules found in all applicable elements. This includes rules specified for the global robots token and rules specified for the accessor's specific product token.

Because all rules specified in this document are restrictive, they are inherently additive. An accessor cannot "opt-out" of a restriction defined in a global 'robots' tag by being mentioned in a specific tag without that restriction. If any applicable tag contains a restrictive rule, that rule MUST be honored.

3.1.3. Robots controls rules

The possible values of the rules are:

* noindex - instructs the parser to not store the served data in its publicly accessible index.

* nosnippet - instructs the parser to not reproduce any stored data as an excerpt snippet.

The values are case insensitive.

Implementors may support other rules as specified in Section 2.2.4 of [ROBOTSTXT].

3.1.4. Caching of values

Implementors SHOULD link the validity of robots-tag rules to the freshness of the associated resource. Rules extracted from an HTTP response header or an HTML meta tag remain in effect until the resource is recrawled or the cached representation expires.

Implementors SHOULD determine the freshness of the rules using standard HTTP cache directives, such as Cache-Control or Expires, as defined in [HTTP-SEMANTICS]. In the absence of explicit cache directives, implementors MAY use heuristics to determine the refresh interval, typically matching the scheduled recrawl frequency of the resource.

If a resource is not recrawled due to a lack of perceived change, the last known robots-tag rules MUST continue to be honored.

4. Security Considerations

The robots-tag is not a substitute for valid content security measures. To control access to the URL paths where the robots-tag appears, service owners SHOULD employ a valid security measure such as HTTP Authentication as defined in [HTTP-SEMANTICS].

The content of the robots-tag header field is not secure, private or integrity-guaranteed, and due caution should be exercised when using it. Use of Transport Layer Security ([TLS]) with HTTP ([HTTP-SEMANTICS]) is currently the only end-to-end way to provide such protection.

In case of a robots-tag specified in a HTML meta element, implementors should consider only the meta elements specified in the head element of the HTML document, which is generally only accessible to the service owner.

Implementors who execute client-side code MUST NOT treat the post-execution DOM state as the exclusive source of truth. Instead, implementors MUST apply the union of all restrictive rules identified in both the initial HTML source and the final DOM state. A restriction present in either state MUST be honored.

To protect against memory overflow attacks, implementers should enforce a limit on how much data they will parse; see [Section 3.1.1] for the lower limit.

5. IANA Considerations

5.1. HTTP Field Name Registration

IANA is requested to register the following HTTP field name in the "Hypertext Transfer Protocol (HTTP) Field Name Registry" according to the procedures defined in Section 18.4 of [HTTP-SEMANTICS]:

Field Name: Robots-Tag

Template: None

Status: permanent

Reference: [This document]

Comments: This field name supersedes the experimental X-Robots-Tag field.

5.2. Robots Control Rules Registry

IANA is requested to create a new "Robots Control Rules" registry. This registry manages the tokens used as parameters within the Robots-Tag HTTP field and, by extension, the content attribute of the robots meta element.

The registration policy for this registry is "IETF Review" or "Expert Review" as defined in [IANA-GUIDELINES].

The initial entries for this registry are:

Rule Name: noindex

Description: Instructs the parser to not store the served data in its publicly accessible index.

Reference: [This document, Section 3.1.3]

Rule Name: nosnippet

Description: Instructs the parser to not reproduce any stored data as an excerpt snippet.

Reference: [This document, Section 3.1.3]

5.3. Deprecation of X-Robots-Tag

The X-Robots-Tag field name was used prior to the standardization of the Robots-Tag field. New implementations MUST NOT use the X- prefix for this field, adhering to the principles in [X-DEPRECATION]. While parsers SHOULD continue to support X-Robots-Tag for backward compatibility with legacy systems, it is formally deprecated by this document.

6. References

6.1. Normative References

[HTTP-CACHING]
Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/rfc/rfc9111>.

[HTTP-SEMANTICS]
Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[IANA-GUIDELINES]
Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[ROBOTSTXT] Koster, M., Illyes, G., Zeller, H., and L. Sassman, "Robots Exclusion Protocol", RFC 9309, DOI 10.17487/RFC9309, September 2022, <https://www.rfc-editor.org/rfc/rfc9309>.

[STRUCTURED-FIELD-VALUES] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/rfc/rfc8941>.

[TLS] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[WEBLINKING] Nottingham, M., "Web Linking", RFC 8288, DOI 10.17487/RFC8288, October 2017, <https://www.rfc-editor.org/rfc/rfc8288>.

[X-DEPRECATION] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating the "X-" Prefix and Similar Constructs in Application Protocols", BCP 178, RFC 6648, DOI 10.17487/RFC6648, June 2012, <https://www.rfc-editor.org/rfc/rfc6648>.

6.2. Informative References

[HTML-META] "HTML Meta Element", 14 April 2026, <https://html.spec.whatwg.org/multipage/semantics.html#the-meta-element>.

[KiB] "KibiByte", 14 October 2022, <https://simple.wikipedia.org/wiki/Kibibyte>.

Acknowledgments

TODO acknowledge.

Author’s Address

Gary Illyes
Google LLC.
Brandschenkestrasse 110
CH-8002 Z端rich
Switzerland
Email: garyillyes@google.com