Workgroup:
Internet Engineering Task Force (IETF)
Internet-Draft: draft-illyes-repext-00
Published: 1 August 2024
Intended Status: Informational
Expires: 2 February 2025
Authors: G. Illyes, Ed.
Google LLC.

# Robots Exclusion Protocol Extension for URI Level Control

## Abstract

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

## About this Document

This note is to be removed before publishing as an RFC. TODO(illyes): add commentable reference on github robotstxt repo.

## 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 2 February 2025.

## Copyright Notice

**Table of Contents**



## 1. Introduction

While the Robots Exclusion Protocol enables service owners to control how, if at all, automated clients known as crawlers may access the URIs on their services as defined by [RFC8288], 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 URI level controls implemented in the response headers, or in case of HTML in the form of a meta tag. This document specifies these control tags, and in case of the response header field, brings it to standards compliance with [RFC9110].

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

### 1.1. Requirements Language

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.

## 2. Specification

### 2.1. Robots control

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

an application level response header.

in case of HTML, one or more meta tags as defined by the HTML specification.

#### 2.1.1. Application Layer Response Header

The application level response header field name is "robots-tag" and contains rules applicable to either all accessors or specifically named ones in the value. For historical reasons, implementors should support the experimental field name also — "x-robots-tag".

The value is a semicolon (";", 0x3B, 0x20) separated list of key-value pairs that represent a comma separated list of rules. The rules are specific to a single product token as defined by [RFC9309] or a global identifier — "*". The global identifier may be omitted. The product token is separated by a "=" from its rules.

Duplicate product tokens must be merged and the rules deduplicated.

```
; key-values definition for the robots-tag response header.
robots-tag = "robots-tag" ":" robots-tag-values
robots-tag-values = *(value ";")
value = ( global-product-token / ( product-token "=" ) ) [rule]
global-product-token = "*" / OWS
product-token =  1*( %x2D / %x41-5A / %x5F / %x61-7A )
rule = "noindex" / "nosnippet"
OWS = *( SP / HTAB )
```

For example, the following response header field specifies "noindex" and "nosnippet" rules for all accessors, however specifies no rules for the product token "ExampleBot":

```
Robots-Tag: *=noindex, nosnippet; ExampleBot=;
```

The global product identifier "*" in the value may be omitted; for example, this field is equivalent to the previous example:

```
Robots-Tag: noindex, nosnippet; ExampleBot=;
```

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

### 2.1.2. HTML meta element

For historical reasons the robots-tag header may be specified by 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:

```
 &lt;meta name="robots" content="noindex"&gt;
 &lt;meta name="examplebot" content="nosnippet"&gt;
```

Multiple robots meta elements may appear in a single HTML document. Requestors must obey the sum of negative rules specific to their product token and the global product token.

## 2.2. Robots control 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. Unsupported rules must be ignored.

Implementors may support other rules as specified in Section 2.2.4 of [RFC9309].

## 2.3. Caching of values

The rules specified for a specific product token must be obeyed until the rules have changed. Implementors MAY use standard cache control as defined in [RFC9110] for caching robots-tag rules. Implementors SHOULD refresh their caches within a reasonable time frame.

# 3. IANA considerations

TODO(illyes):
https://www.rfc-editor.org/rfc/rfc9110.html#name-field-name-registry

# 4. Security considerations

The robots-tag is not a substitute for valid content security measures. To control access to the URI paths in a robots.txt file, users of the protocol should employ a valid security measure relevant

to the application layer on which the robots.txt file is served — for example, in the case of HTTP, HTTP Authentication as defined in [RFC9110].

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 ([RFC9110] and [RFC2817]) 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[a].

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

## 5. References

### 5.1. Normative References


**[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/info/rfc2119>.

**[RFC2817]** Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", RFC 2817, DOI 10.17487/RFC2817, May 2000, <https://www.rfc-editor.org/info/rfc2817>.


### 5.2. Informative References


**[KiB]** "Kibibyte - Simple English Wikipedia, the free encyclopedia", March 2006, <https://simple.wikipedia.org/wiki/Kibibyte>.


## Author's Address


Gary Illyes (editor)
Google LLC.
Brandschenkestrasse 110
CH-8002 Zurich
Switzerland

Email: garyillyes@google.com