

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 October 2026

G. Illyes
Independent
M. Kuehlewind
Ericsson
A. Kohn
Blind Five Year Old
21 April 2026

Crawler best practices
draft-illyes-webbotauth-cbcp-00

Abstract

This document describes best practices for web crawlers.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at
<https://github.com/garyilleyes/cbcp>.

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 23 October 2026.

Copyright Notice

Copyright (c) 2026 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.

Table of Contents

1. Introduction	2
2. Recommended Best Practices	3
2.1. Crawlers MUST respect the Robots Exclusion Protocol	3
2.2. Crawlers MUST be easily identifiable through their user agent string	3
2.3. Crawlers MUST not interfere with the normal operation of a site	4
2.4. Crawlers MUST support caching directives	4
2.5. Crawlers MUST expose the IP ranges they use for crawling	5
2.6. Crawlers MUST explain how the crawled data is used and how the crawler can be blocked	5
3. Conventions and Definitions	5
4. Security Considerations	5
5. IANA Considerations	6
6. References	6
6.1. Normative References	6
6.2. Informative References	7
Acknowledgments	7
Authors' Addresses	7

1. Introduction

Automatic clients, colloquially referred to as crawlers and bots, are used to access web resources, including indexing for search engines or, more recently, training data for new artificial intelligence (AI) applications. As crawling activity increases, automatic clients must behave appropriately and respect the constraints of the resources they access. This includes clearly documenting how they can be identified and how their behavior can be influenced. Therefore, crawler operators are asked to follow the best practices for crawling outlined in this document.

For the purposes of this document, a crawler is an automated HTTP [HTTP-SEMANTICS] client that retrieves resources across one or more web sites without direct human initiation of individual requests. A crawler discovers URIs during retrieval and schedules them for later processing. It relies on algorithmic prioritization and protocol-level instructions such as the Robots Exclusion Protocol [REP] to govern its behavior.

To further assist website owners, it should also be considered to create a central registry where website owners can look up well-behaved crawlers. Note that while self-declared research crawlers, including privacy and malware discovery crawlers, and contractual crawlers are welcome to adopt these practices, due to the nature of their relationship with sites, they may exempt themselves from any of the Crawler Best Practices with a documented rationale.

2. Recommended Best Practices

The following best practices SHOULD be followed and are already applied by a vast majority of large-scale crawlers on the Internet:

1. Crawlers MUST support and respect the Robots Exclusion Protocol.
2. Crawlers MUST be easily identifiable through their user agent string.
3. Crawlers MUST not interfere with the regular operation of a site.
4. Crawlers MUST support caching directives.
5. Crawlers MUST expose the ranges they are crawling from in a standardized format.
6. Crawlers MUST expose a page that explains how the crawling can be blocked, whether the page is rendered, and how the crawled data is used.

2.1. Crawlers MUST respect the Robots Exclusion Protocol

All well behaved-crawlers MUST support the REP as defined in Section 2.2.1 of [REP] to allow site owners to opt out from crawling.

Especially if the website chooses not to use a robots.txt file as defined by the REP, crawlers further need to respect the X-robots-tag in the HTTP header.

2.2. Crawlers MUST be easily identifiable through their user agent string

As outlined in Section 2.2.1 of [REP] (Robots Exclusion Protocol; REP), the HTTP request header User-Agent SHOULD clearly identify the crawler, usually by including a URL that hosts the crawler's description. For example:

User-Agent: Mozilla/5.0 (compatible; ExampleBot/0.1; +https://www.example.com/bot.html)

This is already a widely accepted practice among crawler operators. To remain compliant, crawler operators **MUST** include unique identifiers for their crawlers in the case-insensitive User-Agent header, such as "contains 'googlebot' and 'https://url/...'". Additionally, the name **SHOULD** clearly identify both the crawler owner and its purpose as much as reasonably possible.

2.3. Crawlers **MUST** not interfere with the normal operation of a site

Depending on a site's setup (computing resources and software efficiency) and its size, crawling may slow down the site or even take it offline altogether. Crawler operators **MUST** ensure that their crawlers are equipped with back-out logic that relies on at least the standard signals defined by Section 15.6 of [HTTP-SEMANTICS], preferably also additional heuristics such as a change in the relative response time of the server.

Therefore, crawlers **SHOULD** log already visited URLs, the number of requests sent to each resource, and the respective HTTP status codes in the responses, especially if errors occur, to prevent repeatedly crawling the same source. Using the same data, crawlers **SHOULD**, on a best effort basis, crawl the site at times of the day when the site is estimated to have fewer human visitors.

Generally, crawlers **SHOULD** avoid sending multiple requests to the same resources at the same time and **SHOULD** limit the crawling speed to prevent server overload, if possible, following the limits outlined in [REP]. Additionally, resources **SHOULD NOT** be re-crawled too often. Ideally, crawlers **SHOULD** restrict the depth of crawling and the number of requests per resource to prevent loops.

Crawlers **SHOULD NOT** attempt to bypass authentication or other access restrictions, such as when login is required, CAPTCHAs are in use, or content is behind a paywall, unless explicitly agreed upon with the website owner.

Crawlers **SHOULD** primarily access resources using HTTP GET requests, resorting to other methods (e.g., POST, PUT) only if there is a prior agreement with the publisher or if the publisher's content management system automatically makes those calls through client side code (e.g., JavaScript). Generally, the load caused by executing client side code should be carefully considered or even avoided whenever possible.

2.4. Crawlers **MUST** support caching directives

[HTTP-CACHING] HTTP caching removes the need of repeated access from crawlers to the same URL.

2.5. Crawlers MUST expose the IP ranges they use for crawling

To complement the [REP], crawler operators SHOULD publish the IP ranges they have allocated for crawling in [JAFAR] format, and keep this information reasonably up-to-date, according to the specification.

The resource containing the IP addresses MUST be linked from the page describing the crawler using the client-ranges relation. To facilitate efficient machine discovery. This relation SHOULD be provided via an HTTP Link header [HTTP-LINK] or as a <link> element in the page's HTML metadata section (as defined in [HTML]). For example:

```
<link rel="client-ranges" href="https://example.com/crawlerips.json">
```

2.6. Crawlers MUST explain how the crawled data is used and how the crawler can be blocked

Crawlers MUST be easily identifiable through their user-agent string, and they SHOULD explain how the data they collect will be used. In practice, this is usually done via the documentation page linked in the crawler's user agent. Additionally, the documentation page SHOULD include a contact address for the crawler owner.

The documentation SHOULD also provide an example [REP] file to block the crawler and a method for verifying [REP] files.

If the crawler has exempted itself of these best practices, the documentation page SHOULD describe the reason for that.

All endpoints hosting identification, documentation, and IP range data must be publicly and highly available, and served with minimal latency for programmatic access.

3. 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.

4. Security Considerations

TODO Security

5. IANA Considerations

This document has no IANA actions.

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-LINK]

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

[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>>.

[JAFAR]

Illyes, G., "A JSON-Based Format for Publishing IP Ranges of Automated HTTP Clients", Work in Progress, Internet-Draft, draft-illyes-aipref-jafar-01, 9 April 2026, <<https://datatracker.ietf.org/doc/html/draft-illyes-aipref-jafar-01>>.

[REP]

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>>.

[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>>.

[ROBOTS-TAG]

Illyes, G., "Robots Exclusion Protocol Extension for URL Level Control", Work in Progress, Internet-Draft, draft-

illyes-repext-03, 18 April 2026,
<<https://datatracker.ietf.org/doc/html/draft-illyes-repext-03>>.

6.2. Informative References

[HTML] WHATWG, "HTML", <<https://html.spec.whatwg.org/>>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Gary Illyes
Independent
Email: synack@garyillyes.com

Mirja K^端hlewind
Ericsson
Email: mirja.kuehlewind@ericsson.com

AJ Kohn
Blind Five Year Old
Email: aj@blindfiveyearold.com