Robots.txt 规范

摘要

本文档详细介绍了 Google 处理 robots.txt 文件的方式，这种文件让您可以控制 Google 的网站抓取工具如何抓取可公开访问的网站并将其编入索引。

具体变化

2019 年 7 月 1 日，Google 宣布 (https://webmasters.googleblog.com/2019/07/rep-id.html)将推动 robots.txt 协议成为一项互联网标准 (https://tools.ietf.org/html/draft-koster-rep-00)。这些变化都反映在本文档中。

变化列表

具体变化如下：

- 移除了本文档中的“语言要求”部分，因为语言是与互联网草案相对应的。
- Robots.txt 现在接受所有基于 URI (https://en.wikipedia.org/wiki/Uniform_Resource_Identifier) 的协议。
- Google 会跟踪至少五次重定向。由于尚未提取任何规则，因此系统会跟踪至少五次重定向，如果找不到 robots.txt，Google 会在处理 robots.txt 时将其视为 404 错误。我们不建议用户根据返回 2xx 的 HTML 内容处理 robots.txt 文件的逻辑重定向（框架、JavaScript 或元刷新型重定向），在查找适用的规则时，系统会使用第一个网页的内容。
- 对于 5xx，如果 robots.txt 连续 30 天以上无法访问，系统会使用 robots.txt 的最后一个缓存副本；如果 robots.txt 不可用，Google 会假定没有任何抓取限制。
- Google 会将不成功的请求或不完整的数据视为服务器错误。
- “记录”现在称为“行”或“规则”（视情况而定）。
- Google 不支持处理存在轻微错误或拼写错误的 **`<field>`** 元素（例如，“user-agent”错写成了“useragent”）。
- Google 目前强制执行的文件大小上限为 500 KiB (https://en.wikipedia.org/wiki/Kibibyte)，并会忽略超出该上限的内容。
- 根据 RFC5234 (https://tools.ietf.org/html/rfc5234) 更新了正式语法，使其成为有效的 Augmented Backus-Naur Form (ABNF)，并代替 robots.txt 中的 UTF-8 字符。
- 更新了“组”的定义，使其更简短扼要。添加了空组的示例。

- 移除了对已弃用的 AJAX 抓取机制的引用。

基本定义

定义	
抓取工具	抓取工具是指抓取网站的服务或代理。一般而言，抓取工具会自动以递归方式访问某个主机的已知网址，该主机发布的内容可通过标准的网络浏览器访问。如果抓取工具发现新网址（通过各种途径，例如现有的已抓取网页上的链接或站点地图文件），也会以相同的方式进行抓取。
用户代理	一种用于标识特定抓取工具或一组抓取工具的手段。
指令	robots.txt 文件中规定的某个抓取工具或一组抓取工具适用的一系列准则。
网址	RFC 1738（http://www.ietf.org/rfc/rfc1738.txt) 中定义的统一资源定位器。
Google 专用	这些元素针对的是 Google 的 robots.txt 实现，可能与其他各方无关。

适用性

Google 的所有自动抓取工具均遵循本文档中所述的准则。如果代理代表用户访问网址（例如，为了进行翻译、手动订阅的 Feed、恶意软件分析），则不必遵循这些准则。

文件位置和有效范围

robots.txt 文件必须位于主机的顶级目录中，可通过适当的协议和端口号进行访问。robots.txt 的通用协议都是基于 URI（https://en.wikipedia.org/wiki/Uniform_Resource_Identifier) 的协议，而专用于 Google 搜索（例如，用于抓取网站）的协议为“http”和“https”。按照 http 和 https 协议，使用 HTTP 无条件 GET 请求来抓取 robots.txt 文件。

Google 专用：Google 同样接受和遵循 FTP 网站的 robots.txt 文件。基于 FTP 的 robots.txt 文件可在匿名登录的情况下通过 FTP 协议访问。

robots.txt 文件中列出的指令仅适用于该文件所在的主机、协议和端口号。

网址一样，robots.txt 文件的网址也区分大小写。

有效 robots.txt 网址的示例

robots.txt 网址示例	
`http://example.com/robots.txt`	👍 **适用于：** • `http://example.com/` • `http://example.com/folder/file` 👎 **不适用于：** • `http://other.example.com/` • `https://example.com/` • `http://example.com:8181/` 💡 这属于一般情况。该网址对其他子网域、协议或端口号来说无效；对同一个主机、协议和端口号上的所有子目录中的所有文件有效。
`http://www.example.com/robots.txt`	👍 **适用于：** `http://www.example.com/` 👎 **不适用于：** • `http://example.com/` • `http://shop.www.example.com/` • `http://www.shop.example.com/` 💡 子网域上的 robots.txt 仅对该子网域有效。
`http://example.com/folder/robots.txt`	robots.txt 文件无效。抓取工具不会检查子目录中是否包含 robots.txt 文件。
`http://www.müller.eu/robots.txt`	👍 **适用于：** • `http://www.müller.eu/` • `http://www.xn--mller-kva.eu/` 👎 **不适用于：** `http://www.muller.eu/` 💡 IDN 等同于其对应的 punycode 版本。另请参阅 RFC 3492 (http://www.ietf.org/rfc/rfc3492.txt)。

robots.txt 网址示例

`ftp://example.com/robots.txt`	👍 **适用于：** `ftp://example.com/` 👎 **不适用于：** `http://example.com/` Google 专用：我们会对 FTP 资源使用 robots.txt。
`http://212.96.82.21/robots.txt`	👍 **适用于：** `http://212.96.82.21/` 👎 **不适用于：** `http://example.com/`（即使托管在 212.96.82.21 上） 💡 以 IP 地址作为主机名的 robots.txt 仅在抓取作为主机名的该 IP 地址时有效。此类 robots.txt 并不会自动对该 IP 地址上托管的所有网站有效（但该文件可能是共享的，在此情况下，该文件也可以在共享主机名下使用）。
`http://example.com:80/robots.txt`	👍 **适用于：** • `http://example.com:80/` • `http://example.com/` 👎 **不适用于：** `http://example.com:81/` 💡 标准端口号（http 为 80；https 为 443；ftp 为 21）等同于其默认的主机名。另请参阅 **`[portnumbers]`**。
`http://example.com:8181/robots.txt`	👍 **适用于：** `http://example.com:8181/` 👎 **不适用于：** `http://example.com/` 💡 非标准端口号上的 robots.txt 文件仅对通过这些端口号提供的内容有效。

处理 HTTP 结果代码

一般情况下，robots.txt 文件会出现三种不同的抓取结果：

- 全部允许：所有内容均可抓取。
- 全部禁止：所有内容均不能抓取。
- 有条件地允许：robots.txt 中的指令决定是否可以抓取某些内容。

处理 **HTTP** 结果代码	
2xx（成功）	HTTP 结果代码，表示成功的“有条件地允许”抓取结果。
3xx（重定向）	Google 会跟踪至少五次重定向（如适用于 HTTP/1.0 的 RFC 1945 (http://www.ietf.org/rfc/rfc1945.txt) 所规定），然后便会停止，并将其处理为 404 错误。我们不建议用户处理指向禁止网址的 robots.txt 重定向；由于尚未提取任何规则，因此系统会跟踪至少五次重定向，如果找不到 robots.txt，Google 会在处理 robots.txt 时将其视为 404 错误。我们不建议用户根据返回 2xx 的 HTML 内容处理 robots.txt 文件的逻辑重定向（框架、JavaScript 或元刷新型重定向），在查找适用的规则时，系统会使用第一个网页的内容。
4xx（客户端错误）	系统对所有 4xx 错误都采用同一种处理方式，并且假定不存在有效的 robots.txt 文件。Google 假定不存在任何限制。这表示抓取时“全部允许”。 ★ 包括 401“未授权”和 403“禁止访问”HTTP 结果代码。
5xx（服务器错误）	我们将服务器错误视作会导致抓取作业“全部禁止”的临时性错误。系统会再次尝试发送该请求，直到获得非服务器错误的 HTTP 结果代码。503（服务不可用）错误会导致非常频繁的重试操作。如果 robots.txt 连续 30 天以上无法访问，系统会使用 robots.txt 的最后一个缓存副本；如果 robots.txt 不可用，Google 会假定没有任何抓取限制。要暂停抓取，我们建议您提供 503 HTTP 结果代码。 Google 专用：如果我们能够确定，某网站因为配置不正确而在缺少网页时返回 5xx 错误而不是 404 错误，就会将该网站的 5xx 错误作为 404 错误处理。
请求不成功或数据不完整	系统会将因 DNS 或网络问题（超时、响应无效、重置或断开连接、HTTP 组块错误等）而无法抓取的 robots.txt 文件的处理视为服务器错误 (#server-error)。
缓存	一般情况下，robots.txt 内容最多可缓存 24 小时，但在无法刷新缓存版本的情况下（例如，出现超时或 5xx 错误时），缓存时间可能会延长。缓存的响应可由各种不同的抓取工具共享。Google 会根据 max-age Cache-Control (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9.3) HTTP 标头延长或缩短缓存生命周期。

文件格式

预期的文件格式是 UTF-8 (http://en.wikipedia.org/wiki/UTF-8) 编码的纯文本。文件包含由 CR、CR/LF 或 LF 分隔的多个行。

系统将只考虑有效的行，而忽略其他所有内容。例如，如果获得的文档为 HTML 网页，则系统只会考虑网页中有效的文本行，而忽略其他内容，并且既不显示警告也不报告错误。

如果因为使用某种字符编码而引入了不属于 UTF-8 子集的字符，则可能导致文件内容解析错误。

系统会忽略 robots.txt 文件开头可选的 Unicode BOM（http://en.wikipedia.org/wiki/Byte_order_mark）（字节顺序标记）。

每个有效行均由一个字段、一个冒号和一个值组成。空格是可选的（但建议使用空格以提高可读性）。您可以使用“#”字符在文件中的任何位置添加注释，系统会将所有位于注释开头和行结尾之间的内容视为注释，并且忽略这部分内容。常见格式为 **<field>:<value><#optional-comment>**。系统会忽略行开头和结尾的空格。

<field> 元素不区分大小写。<value> 元素可能会区分大小写，具体取决于 <field> 元素。

我们不支持处理存在轻微错误或拼写错误的 **<field>** 元素（例如，“user-agent”错写成了“useragent”）。

每个抓取工具都可以单独设定文件大小的上限，并忽略超过该上限的内容。Google 目前强制执行的文件大小上限为 500 KiB（https://en.wikipedia.org/wiki/Kibibyte）。要减小 robots.txt 文件的大小，请将会导致 robots.txt 文件过大的指令整合在一起。例如，将已排除的内容放在一个单独的目录中。

正式语法/定义

这是 Augmented Backus-Naur Form (ABNF) 说明，如 RFC 5234（https://www.ietf.org/rfc/rfc5234.txt) 中所述

```
robotstxt = *(group / emptyline)
group = startgroupline                      ; We start with a user-agent
        *(startgroupline / emptyline)       ; ... and possibly more user-agents
        *(rule / emptyline)                 ; followed by rules relevant for UAs

startgroupline = *WS "user-agent" *WS ":" *WS product-token EOL

rule = *WS ("allow" / "disallow") *WS ":" *WS (path-pattern / empty-pattern) EOL

; parser implementors: add additional lines you need (for example, Sitemaps), and
; be lenient when reading lines that don’t conform. Apply Postel’s law.

product-token = identifier / "*"
path-pattern = "/" *(UTF8-char-noctl)    ; valid URI path pattern; see 3.2.2
empty-pattern = *WS

identifier = 1*(%x2d / %x41-5a / %x5f / %x61-7a)
comment = "#" *(UTF8-char-noctl / WS / "#")
emptyline = EOL
EOL = *WS [comment] NL          ; end-of-line may have optional trailing comment
NL = %x0D / %x0A / %x0D.0A
WS = %x20 / %x09
```

```
; UTF8 derived from RFC3629, but excluding control characters
UTF8-char-noctl = UTF8-1-noctl / UTF8-2 / UTF8-3 / UTF8-4
UTF8-1-noctl    = %x21 / %x22 / %x24-7F  ; excluding control, space, '#'
UTF8-2          = %xC2-DF UTF8-tail
UTF8-3          = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2( UTF8-tail ) /
                  %xED %x80-9F UTF8-tail / %xEE-EF 2( UTF8-tail )
UTF8-4          = %xF0 %x90-BF 2( UTF8-tail ) / %xF1-F3 3( UTF8-tail ) /
                  %xF4 %x80-8F 2( UTF8-tail )
UTF8-tail       = %x80-BF
```

行和规则分组

一个或多个用户代理行，后跟一个或多个规则。组以用户代理行或文件末尾终止。最后一个组可能没有规则，这意味着它暗含的意思是允许所有内容。

示例组：

```
user-agent: a
disallow: /c

user-agent: b
disallow: /d

user-agent: e
user-agent: f
disallow: /g

user-agent: h
```

该示例中指定了四个不同的组，一个针对“a”，一个针对“b”，还有一个同时针对“e”和“f”。除了最后一个组外，每个组都有各自的组成员行。最后一个组是空的。注意：您可以选择使用空格和空白行来提高可读性。

用户代理的优先顺序

对于某个抓取工具而言，只有一个组是有效的。抓取工具必须查找最具体的匹配用户代理，从而确定正确的行组。其他所有组则一律忽略。用户代理区分大小写。所有非匹配文本都会被忽略（例如，**googlebot/1.2** 和 **googlebot*** 均等同于 **googlebot**）。这与 robots.txt 文件中的组顺序无关。

如果特定用户代理有多个组，则这些组中适用于特定用户代理的所有规则会合并在一起。

示例

以下面的 robots.txt 文件为例：

```
user-agent: googlebot-news
(group 1)

user-agent: *
(group 2)

user-agent: googlebot
(group 3)
```

以下为抓取工具选择相关组的方法：

每个抓取工具追踪的组	
Googlebot 新闻	追踪的组是组 1。仅追踪最具体的组，而忽略其他所有组。
Googlebot（网络）	追踪的组是组 3。
Googlebot 图片	追踪的组是组 3。没有具体的 **googlebot-images** 组，因此将追踪更宽泛的组。
Googlebot 新闻（抓取图片时）	追踪的组是组 1。这些图片由“Googlebot 新闻”抓取和使用，因此将仅追踪“Googlebot 新闻”组。
其他机器人（网络）	追踪的组是组 2。
其他机器人（新闻）	追踪的组是组 2。即使有相关抓取工具的对应条目，也只有在明确匹配时才会有效。

另请参阅 Google 的抓取工具和用户代理字符串
(https://support.google.com/webmasters/answer/1061943?hl=zh-cn)

组成员规则

本部分仅说明标准的组成员规则。对于抓取工具，这些规则也称为“指令”。这些指令以 **directive: [path]** 的形式指定，其中 **[path]** 可选。默认情况下，指定的抓取工具没有抓取限制。没有 **[path]** 的指令会被忽略。

如果指定了 **[path]** 值，该路径值将被视作 robots.txt 文件抓取网站的根目录的相对路径（使用相同的协议、端口号、主机和域名）。路径值必须以“/”开头，表示根目录。路径区分大小写。有关详情，请参阅下面的“基于路径值的网址匹配”部分。

disallow

disallow 指令指定相应抓取工具不能访问的路径。如果未指定路径，该指令将被忽略。

用法：

```
disallow: [path]
```

allow

allow 指令指定相应抓取工具可以访问的路径。如果未指定路径，该指令将被忽略。

用法：

```
allow: [path]
```

基于路径值的网址匹配

以路径值为基础确定某项规则是否适用于网站上的特定网址。不使用通配符时，路径可用于匹配网址的开头（以及以相同路径开头的任何有效网址）。路径中的非 7 位 ASCII 字符可以作为 UTF-8 字符添加，也可以按照 RFC 3986 (http://www.ietf.org/rfc/rfc3986.txt) 作为百分号转义的 UTF-8 编码字符添加。

对于路径值，Google、Bing 和其他主要搜索引擎支持有限形式的“通配符”。这些通配符包括：

- * 表示任何有效字符的 0 个或多个个案。
- $ 表示网址结束。

路径匹配示例	
`/`	匹配根目录以及任何下级网址
`/*`	等同于 `/`。结尾的通配符会被忽略。
`/fish`	👍 **匹配项：** • `/fish` • `/fish.html` • `/fish/salmon.html` • `/fishheads` • `/fishheads/yummy.html` • `/fish.php?id=anything` 👎 **不匹配项：** • `/Fish.asp` • `/catfish` • `/?id=fish` ★ 注意：比对时区分大小写。
`/fish*`	等同于 `/fish`。结尾的通配符会被忽略。 👍 **匹配项：** • `/fish` • `/fish.html` • `/fish/salmon.html` • `/fishheads` • `/fishheads/yummy.html` • `/fish.php?id=anything` 👎 **不匹配项：** • `/Fish.asp` • `/catfish` • `/?id=fish`

路径匹配示例	
`/fish/`	结尾的斜杠表示此项与此文件夹中的任何内容均匹配。 👍 **匹配项：** • `/fish/` • `/fish/?id=anything` • `/fish/salmon.htm` 👎 **不匹配项：** • `/fish` • `/fish.html` • `/Fish/Salmon.asp`
`/*.php`	👍 **匹配项：** • `/filename.php` • `/folder/filename.php` • `/folder/filename.php?parameters` • `/folder/any.php.file.html` • `/filename.php/` 👎 **不匹配项：** • `/`（即使其映射到 /index.php） • `/windows.PHP`
`/*.php$`	👍 **匹配项：** • `/filename.php` • `/folder/filename.php` 👎 **不匹配项：** • `/filename.php?parameters` • `/filename.php/` • `/filename.php5` • `/windows.PHP`

路径匹配示例	
`/fish*.php`	匹配项： • `/fish.php` • `/fishheads/catfish.php?parameters` 不匹配项：`/Fish.PHP`

Google 支持的非组成员行

Google、Bing 和其他主要搜索引擎支持 **sitemap**（如 sitemaps.org（http://sitemaps.org) 所定义）。

用法：

```
sitemap: [absoluteURL]
```

[absoluteURL] 指向站点地图、站点地图索引文件或等效网址。网址不需要与 robots.txt 文件位于同一主机上。**sitemap** 条目可以有多个。作为非组成员行，它们不依赖于任何特定的用户代理，只要未加禁止，所有抓取工具都可以追踪它们。

组成员行的优先顺序

在组成员一级，尤其是对于 **allow** 和 **disallow** 指令，最具体的规则（根据 **[path]** 条目的长度，长度越短，越不具体）优先级最高。如果规则（包括使用通配符的规则）存在冲突，系统将使用限制性最弱的规则。

示例情况	
`http://example.com/page`	**allow**: **/p** **disallow**: **/** 认定结果：**allow**

示例情况	

`http://example.com/folder/page`	`allow`: `/folder`
	`disallow`: `/folder`
	
	认定结果：`allow`

`http://example.com/page.htm`	`allow`: `/page`
	`disallow`: `/*.htm`
	认定结果：`undefined`

`http://example.com/`	`allow`: `/$`
	`disallow`: `/`
	认定结果：`allow`

`http://example.com/page.htm`	`allow`: `/$`
	`disallow`: `/`
	认定结果：`disallow`

Last updated 2020-02-21.