Stephan Czysch

CRAWLER MIT ROBOTS.TXT IN DIE SCHRANKEN WEISEN

Für Crawler gilt: Alles, was nicht verboten ist, ist erlaubt. Das Crawling erlauben? Nicht notwendig! Die Indexierung erlauben? Ebenfalls nicht notwendig! Nur dann, wenn Seiten (URLs) nicht gecrawlt oder nicht indexiert werden sollen, dann ist ein Eingreifen mittels Disallow-Anweisung oder Noindex-Tag erforderlich.
Von der Prozessseite betrachtet findet das Crawling vor der Indexierung statt. Entsprechend kann nur für gecrawlte Dokumente eine Noindex-Angabe gefunden und beachtet werden. Über die robots.txt-Datei wird das Crawling gesteuert – das macht die robots.txt zu einem sehr mächtigen und häufig unterschätzten Tool. Wie diese kleine Datei eingesetzt werden kann, erklärt Stephan Czysch in diesem Artikel.

DER AUTOR

Stephan unterstützt als Sparringspartner und durch Workshops Inhouse-SEO-Teams dabei, mehr zu erreichen und bessere Ergebnisse zu erzielen.
Er beschäftigt sich intensiv mit datengetriebenem Online-Marketing und hat gerade die Beta-Phase seines Google-Search-Console-Tools searchanalyzer.io gestartet sowie das Buch „Local SEO verständlich erklärt“ veröffentlicht.

Egal, ob man sie Crawler, Spider, Robots oder im Fall der Suchmaschine aus Mountain View Googlebot nennt: Die kleinen Helferlein für die Durchsuchbarkeit des Webs sind ziemlich gierig. Jede einem Crawler bekannte und nicht vom Crawling via robots.txt ausgeschlossene Adresse kann von diesen kleinen Programmen besucht und anschließend indexiert werden. Indexiert zumindest dann, wenn die Seite erfolgreich abgerufen werden kann (also den Statuscode 200 auf die Anfrage zurückliefert) und der Webmaster die Indexierung nicht durch eine Noindex-Anweisung oder ein auf eine andere Adresse zeigendes Canonical-Tag verhindert – oder im Fall des Canonical-Tags eher zu verhindern versucht. Denn ein Canonical ist nur ein Hinweis, den Suchmaschinen besonders dann ignorieren, wenn sich die Seiteninhalte zwischen den Adressen in Beziehung gesetzter Dokumente (deutlich) unterscheiden.

Natürlich greifen auch aufseiten der Suchmaschine ein paar Logiken, die über die Indexierung einer Adresse entscheiden, da diese vor allem auf der Suche nach im Web einzigartigen und hochwertigen Dokumenten sind. Entsprechend kommt es häufig vor, dass Seiten nicht den Weg in den Index finden, obwohl der Webmaster dies zumindest nicht unterbindet.

Dabei läuft vereinfacht gesagt folgender Prozess ab:

1. Durch einen Verweis von einer bekannten Quelle lernt eine Suchmaschine eine neue URL (= Adresse) kennen. In der Regel ist der Verweis ein interner Link oder die Einreichung einer Adresse via XML-Sitemap. Doch auch durch eine direkte Anmeldung, vor allem über die Google Search Console (oder allgemeiner: die Webmaster-Tools des Suchmaschinenkonzerns), kann eine Adresse bekannt gemacht werden. Ein sogenannter Scheduler plant, wann die URL gecrawlt wird (in der Google Search Console wird dies „Crawling-Warteschlange“ genannt).
2. Wenn die URL an der Reihe ist, wird die Adresse auf einen Crawling-Ausschluss in der robots.txt überprüft. Liegt kein Ausschluss vor, dann wird die Adresse besucht.
3. Sofern die Seite indexiert werden darf und Qualitätstests der Suchmaschine besteht, dann kann sie in den Index aufgenommen werden.

Das Crawling kommt dabei vor der Indexierung: Im Zuge des Crawlings wird der Quelltext einer Seite, genauer der gerenderte Quelltext, eingelesen. Gefundene Indexierungsausschlüsse über ein Noindex als Meta-Robots- oder X-Robots-Angabe sorgen dann dafür, dass die URL auf Wunsch des Webmasters nicht indexiert wird.

Doch warum tauchen Seiten im Google-Index auf, die per robots.txt gesperrt sind? Die Antwort ist aufgrund der oben beschriebenen Abfolge klar: Die URL ist der Suchmaschine

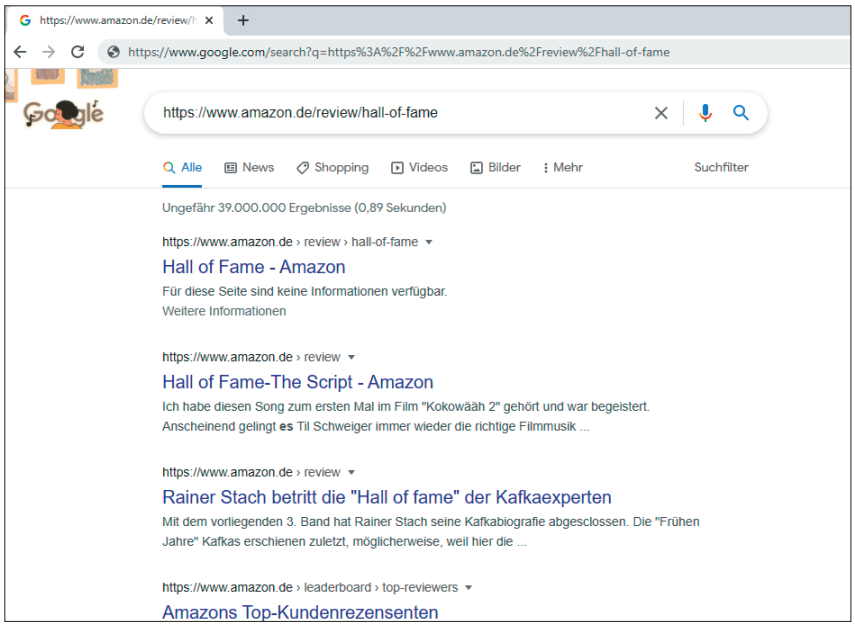

Abb. 1: Da der Zugriff auf die URL via robots.txt blockiert wird, zeigt Google anstelle der Meta-Description nur „Für diese Seite sind keine Informationen verfügbar“ an

durch einen Verweis bekannt, der Seiteninhalt darf aber nicht ausgelesen werden. Entsprechend können alle im Quelltext zu findenden Anweisungen nicht „gelesen“ werden. Dasselbe gilt für den HTTP-Header, über den beispielsweise der Statuscode übermittelt wird. Folglich kennt die Suchmaschine bis auf die Adresse und häufig Ankertexte nichts über die Seite – und entscheidet sich manchmal für deren Indexierung – so zu sehen in Abbildung 1. Es ist möglich, dass eine gesperrte URL eine Fehlerseite anzeigt oder weiterleitet. Doch das weiß die Suchmaschine durch das Crawling-Verbot nicht.

Ist das schlimm? Meiner persönlichen Einschätzung nach nicht, doch hier scheiden sich die SEO-Geister. Viele SEOs meiden ein Disallow mit dem Verweis auf das Thema „Index-Hygiene“ beziehungsweise „Index-Optimierung“ – es sollen nur die Seiten in den Google-Index, die auch eine (relevante) Suchnachfrage bedienen können. Im Sinne der „Index-Hygiene“ gilt folglich das Mantra: „Weniger ist mehr.“ Dem stimme ich grundsätzlich zu, bin allerdings ganz persönlich der Meinung: „Ein Disallow zur rechten Zeit erspart viel Unannehmlichkeit.“ Warum? Einfach dranbleiben!

Wer die Indexierung der eigenen Website also richtig unter Kontrolle haben möchte, der greift auf das Canonical-Tag oder ein beherztes Noindex zurück. Suchmaschinen dürfen die Adressen also crawlen, aber eben nicht indexieren. Die bessere Wahl? Ansichtssache :)

Welche Angaben versteht Google in der robots.txt?

Vorneweg: Zur robots.txt gibt es unter *http://einfach.st/robots44* ein eigenes Kapitel in der Google-Hilfe. Diese Hilfe-Artikel sollten als Primärquelle angesehen werden – besonders die englische Version, da durch die Übersetzung schon mal Unklarheiten oder Fehler entstehen können.

Ganz grundsätzlich versteht Google diese Angaben innerhalb der robots.txt:

» user-agent: Definiert, für welchen Crawler die Regeln gelten sollen. Der User-Agent ist dabei vereinfacht gesagt die Kennung des Crawlers.
» allow: Gibt an, welche URLs gecrawlt werden dürfen. Ein Allow ist eigentlich nur dann notwendig, wenn eine Disallow-Angabe für bestimmte Adressen überschrieben werden soll.
» disallow: Hiermit werden URLs vom Crawling ausgeschlossen.
» sitemap: um eine oder mehrere XML-Sitemap-Dateien zu finden. Die Sitemap muss sich dabei nicht auf demselben Hostnamen befinden wie die aktuell betrachtete robots.txt.

Mit Blick auf den User-Agent ist es wichtig zu wissen, dass es nicht „den einen Googlebot“ gibt, sondern besonders für unterschiedliche Inhaltstypen (z. B. Bilder) oder Suchräume (z. B. Google News) eigene Crawler mit eigener Nutzerkennung (= User-Agent) aktiv sind. Die Übersicht der aktuell 18 von Google genutzten Crawler und User-Agents ist unter *https://developers.google.com/search/docs/advanced/crawling/overview-google-crawlers?hl=de* zu finden. Und damit wir uns nicht falsch verstehen: Es gibt nicht nur einen „Googlebot-News“, sondern ganz viele, die alle dieselbe Kennung verwenden und sich zeitgleich an unterschiedlichen Ecken des Webs aufhalten.

Und noch etwas ist wichtig: Auch bei der robots.txt wird zwischen Groß- und Kleinschreibung unterschieden! Eine Angabe „disallow: /Hallo“- gilt nur für Adressen mit einem großen „H“ und einem anschließend folgenden kleingeschriebenen „allo“ direkt nach einem Verzeichnis. Hinter dem Hallo dürfen sich dabei beliebig viel weitere Zeichen anschließen – folglich ist auch ein /Hallowelt durch die Angabe blockiert.

TIPP

Im Rahmen der SEOkomm haben Jan-Peter Ruhso vom Logfile-Tool crawlOPTIMIZER und Darius Erdt von der Digital-Agentur Dept den Crawling- und Indexierungsprozess sehr genau beschrieben. Die schematische Darstellung des Prozesses kann unter *https://www.crawloptimizer.com/google-indexierungsprozess/* angesehen werden.

robots.txt-Angaben mit Platzhaltern spezifizieren

Innerhalb der robots.txt kann mit den Platzhaltern * und $ gearbeitet werden. Dabei steht * für 0 oder mehr Instanzen von beliebigen Zeichen, während durch $ das Ende einer URL markiert wird. Dadurch lassen sich deutlich spezifischere Angaben schreiben. Beispielsweise können nur Adressen vom Crawling ausgeschlossen werden, die auf .php5 enden (mittels disallow: /*.php5$).

Wer einige Beispiele rund um die Platzhalter sowie die Groß- und Kleinschreibung sucht, der sollte unter *http://einfach.st/robots55* vorbeischauen.

Die Krux mit der Spezifität

disallow: ist eine mächtige Angabe, da es eben den Zugriff des Crawlers auf Adressen unterbindet. Eigentlich ist hinsichtlich des Crawlings alles erlaubt, es sei denn, ein Disallow ist gesetzt und wird nicht über eine spezifischere Angabe überschrieben. Und genau hier wird es manchmal unübersichtlich.

Die folgenden Beispiele sind von *http://einfach.st/robotsrules* übernommen worden.

Beispiel 1

In der robots.txt sind folgende Angaben zu finden:

» *allow: /p*
» *disallow: /*

Und es geht um die Frage, ob *http://example.com/page* gecrawlt werden darf.

disallow: / definiert, dass grundsätzlich keine Adresse der Website gecrawlt werden darf. Die Regel allow: /p überschreibt das allerdings für Adressen, die ein /p beinhalten. Welche Regel kommt also zur Anwendung?

In diesem Fall darf die Adresse gecrawlt werden, da allow: /p die spezifischere Angabe ist.

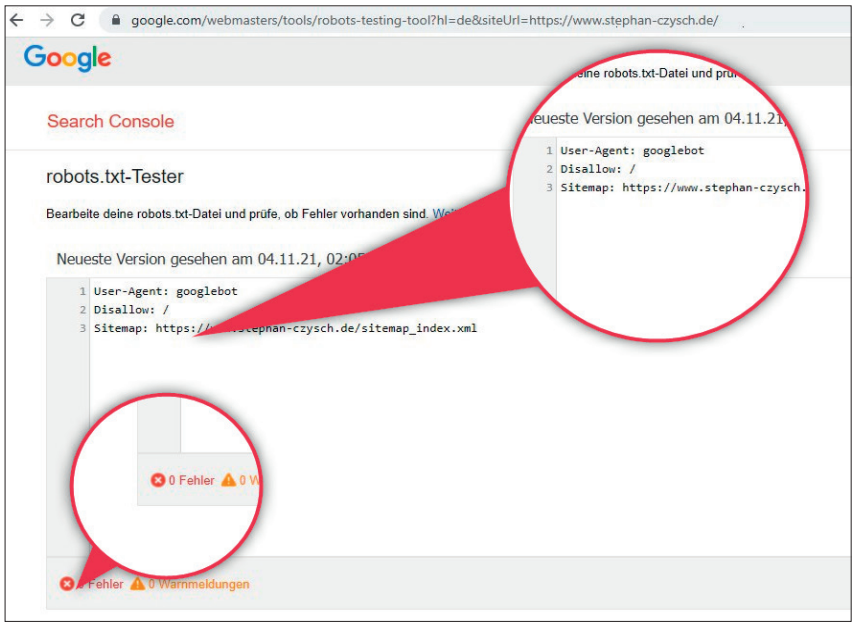

Abb. 2: Mit dem robots.txt-Tester in der Google Search Console können u. a. einzelne Adressen auf einen Crawling-Ausschluss getestet werden

Beispiel 2

Dieses Mal sind folgende Angaben in der robots.txt zu finden:

» *allow: /folder*
» *disallow: /folder*

Das mag erst mal komisch aussehen, kommt aber durchaus vor, besonders bei robots.txt-Dateien mit vielen Angaben. Doch darf *http://example.com/folder/page* gecrawlt werden?

Die Antwort ist ja, da Google bei widersprüchlichen Regeln die am wenigsten restriktive Angabe verwendet.

Beispiel 3

Auch im nachfolgenden Beispiel geht es um das Zusammenspiel von Allow und Disallow. In der robots.txt ist definiert:

» *allow: /page*
» *disallow: /*.htm*

Darf auf die Seite *http://example.com/page.htm* zugegriffen werden?

In dieser Konstellation gewinnt die Angabe disallow:, die Adresse darf also nicht gecrawlt werden. Der Hintergrund: * steht für beliebig viele Zeichen und besagt, dass Adressen mit beliebig vielen und beliebigen Zeichen von Crawling ausgeschlossen sind, wenn diese ein .htm beinhalten. Dadurch stimmt diese Angabe für mehr Zeichen in der Adresse überein und ist damit spezifischer.

Beispiel 4

Definiert ist:

» allow: /page
» disallow: /*.ph

Und es geht um die Adresse *http://example.com/page.php5*. Darf sie gecrawlt werden?

Dieses Mal ist das Crawling erlaubt, da *.ph hier unspezifischer ist als die Angabe allow: – entsprechend wird die am wenigsten restriktive Angabe verwendet.

Beispiel 5

Langsam wirst du mit dem Thema warm, oder? Folgende Anweisungen sind in der robots.txt zu finden:

» *allow: /$*
» *disallow: /*

Darf auf die Adresse *http://example.com/* zugegriffen werden? Die Angabe allow: /$ definiert durch das $-Zeichen eine Freigabe der Startseite, das Disallow verbietet aber grundsätzlich das Crawling.

Für die Startseite gilt die Crawling-Freigabe, da die Anweisung allow: spezifischer ist.

Beispiel 6

Die Angaben aus dem vorherigen Beispiel kommen wieder zum Einsatz, aber eine andere URL muss auf die Crawling-Anweisung überprüft werden. In der robots.txt ist definiert:

» *allow: /$*
» *disallow: /*

Und es geht darum, ob *http://example.com/page.htm* gecrawlt werden darf.

Sie ahnen es vermutlich: allow: /$ bezieht sich nur auf die Startseite, entsprechend darf die Unterseite nicht gecrawlt werden.

Welche Regelgruppe kommt zum Einsatz?

Allow und Disallow sorgen definitiv regelmäßig für Stirnrunzeln. Zumindest bei den User-Agents ist es deutlich einfacher, da hier immer die spezifischste Gruppe zur Anwendung kommt. Es werden also nicht Angaben für User-Agent: * und User-Agent: Googlebot zusammengefasst, sondern Googlebot beachtet nur die Regeln, mit denen er exakt angesprochen wird.

Angaben wie

» user-agent: *
» disallow: /page
» user-agent: googlebot
» disallow:

führen dazu, dass Google die Adresse /page crawlen darf.

Wie kann die robots.txt überprüft werden?

Bereits bei zwei sich auf eine Adresse beziehenden Regeln kann die Frage „Was greift jetzt noch mal?“ herausfordernd sein. In der Praxis sind solche Konflikte zwar selten, aber es gibt sie – und manchmal sind sogar noch weitere Regeln mit in der Verlosung.

Um die robots.txt zu überprüfen, ist das entsprechende Test-Tool in der Google Search Console die passende Anlaufstelle. Das Tool steht unter *http://einfach.st/robotstesting* sowie in der Google Search Console über „Vorherige Tools und Berichte => Weitere Informationen“ zur Verfügung.

Das Tool kann sowohl für die Prüfung einzelner Adressen für die aktuelle robots.txt als auch zum Testen möglicher neuer Regeln verwendet werden. Dazu werden die modifizierten Angaben einfach im obigen Feld eingetragen und gegen den gewünschten User-Agent (Auswahl unten rechts) überprüft. Der robots.txt-Tester kann zudem genutzt werden, um Google via

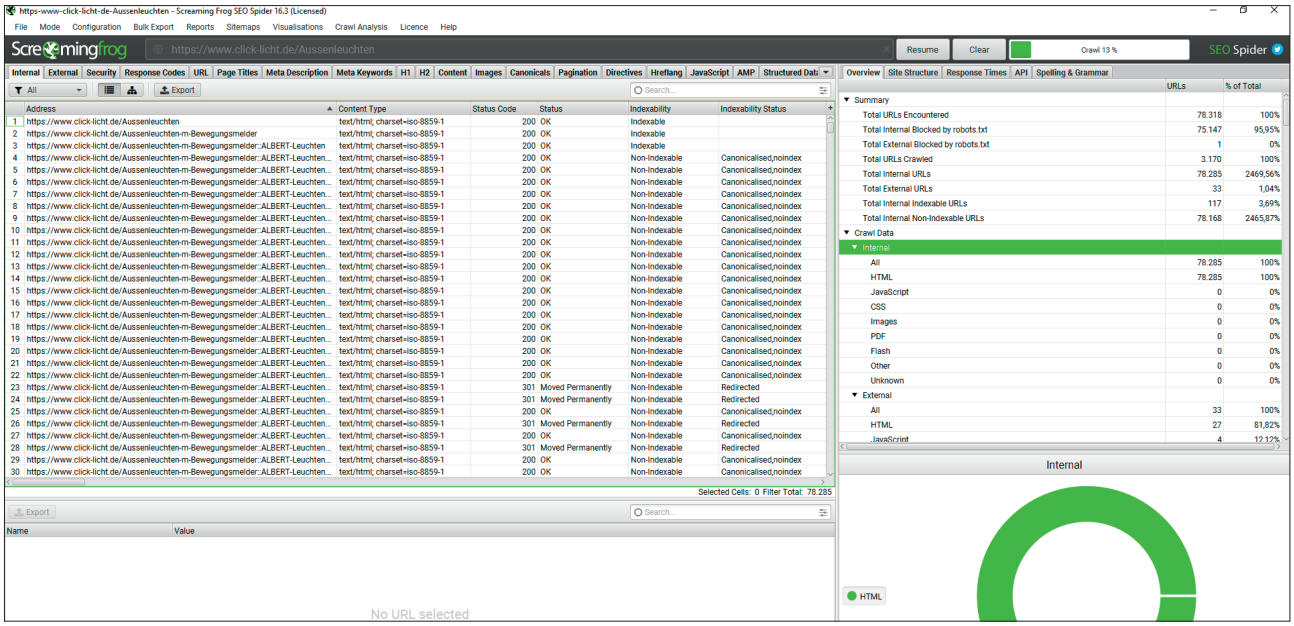

Abb. 3: Durch die Kombination der Filter entstehen unheimlich viele Adressen; in diesem Crawl sind nach 78.258 gecrawlten URLs noch 505.008 Adressen – und die Warteschlange wächst (noch)

„Senden“ über eine neue Version der robots.txt zu informieren. Dazu muss natürlich die robots.txt auf dem Webserver angepasst werden (und nicht nur im Test-Tool).

Wer nicht nur eine, sondern gleich mehrere Adressen auf einen Crawling-Ausschluss überprüfen möchte, der kann einen Blick auf *http://einfach.st/githubrobots* werfen. Hier stellt Google seine „Interpretation“ der robots.txt-Regeln zur Verfügung. Alternativ kann über Crawler wie den Screaming Frog eine robots.txt-Prüfung vorgenommen werden. Der Screaming Frog erlaubt dabei auch ein Überschreiben der robots.txt durch eigene Regeln.

Ein kleiner Tipp für eine schnelle und sinnvolle Analyse: Aus der Google Search Console oder der Webanalyse die organischen Einstiegsseiten nehmen und diese auf einen Crawling-Ausschluss prüfen. Denn auch per robots.txt geprüfte Seiten können (für wenig kompetitive Suchanfragen) in den Suchergebnissen auftauchen und Klicks erzielen. Womöglich landen Nutzer über gesperrte URLs von Google kommend auf Fehlerseiten?! Am schnellsten kann diese Überprüfung im List-Mode des Screaming Frog mit aktivierter Überprüfung der robots.txt (via Crawling => robots.txt => Settings) durchgeführt werden.

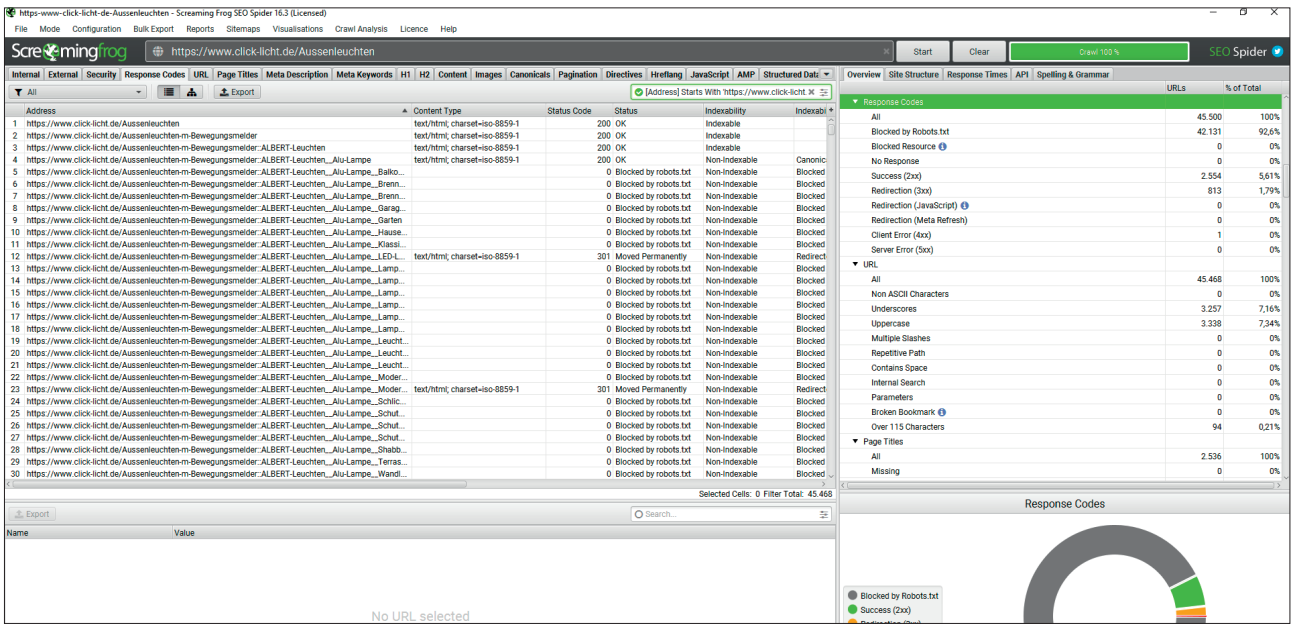

Abb. 4: Wird die robots.txt beachtet, dann sind deutlich weniger Adressen zu finden; der Großteil der Adressen ist durch ein Disallow gesperrt – deshalb konnten nur 2.536 crawlbare URLs gefunden werden

Warum kann ein Disallow aus meiner Sicht für große Seiten sinnvoll sein?

Du hast das „große“ Seiten nicht überlesen, oder? Mit großen Websites meine ich Webauftritte, die mehr als 1.000.000 URLs auf der Website haben, ganz egal, wie diese von Suchmaschinen behandelt werden sollen. Besonders Online-Shops mit vielen Produkten und Filtern gelangen sehr schnell an diese Grenze, da sich die angebotenen Filter meistens miteinander kombinieren lassen.

Ein Beispiel gefällig? Im Lampen-Shop *click-licht.de* ist genau das der Fall. So findet man:

» *https://www.click-licht.de/Aussenleuchten* als Einstieg für Außenleuchten
» *https://www.click-licht.de/Aussenleuchten-m-Bewegungsmelder* als Einstieg für Außenleuchten mit Bewegungsmelder
» *https://www.click-licht.de/Aussenleuchten-m-Bewegungsmelder::ALBERT-Leuchten* als Einstieg für Außenleuchten mit Bewegungsmelder der Marke Albert

Der in Abbildung 3 zu sehende Crawl ist unvollständig, doch nach 78.258 gecrawlten Adressen sind bereits 505.008 Adressen in der Warteschleife – und mit jeder neuen URL werden momentan noch weitere einzigartige Adressen gefunden. Denn die Adresse einer Seite für „Außenleuchten mit Bewegungsmelder aus Aluminium in Schwarz“ lässt sich nur erreichen, wenn bereits eine URL gecrawlt wurde, die alle bis auf eine Filtereigenschaft anzeigt oder mindestens eine zusätzliche Eigenschaft darstellt. Das trifft z. B. auf „Schwarze Außenleuchten mit Bewegungsmelder“ oder „Schwarze Außenleuchten mit Bewegungsmelder aus Aluminium mit modernem Stil“ zu. Wird entweder ein Filter hinzugefügt (in diesem Fall: Material Aluminium) oder entfernt (hier: moderner Stil), dann wird die entsprechende Zielseite erreichbar.

Um den Crawler nicht zu überfordern, hat sich der Shop für einen Crawling-Ausschluss mittels der robots.txt entschieden: Durch diverse Regeln wird der Crawler davon abgehalten, sehr tief in die Facettierung „reinzulaufen“. Der Effekt: Bei Beachtung der robots.txt-Regeln findet ein weiterer Crawl „nur“ 45.468 Adressen innerhalb der Außenleuchten-Kategorie. Für die Adleraugen: Die Anzahl der

gecrawlten Adressen in der Abbildung ist höher, da einige externe Seiten abgerufen wurden.

Von den knapp 45.000 bekannten Adressen sind nur 2.536 crawlbar und wiederum nur 123 URLs können indexiert werden, da sie weder per Canonical-Tag noch per Noindex aus dem Index gehalten werden.

Ganz perfekt ist dieses Vorgehen nicht, da Google satte 42.130 Adressen über interne Verlinkungen zu sehen bekommt, die vom Crawling ausgeschlossen werden. Doch was passiert mit diesen Adressen?

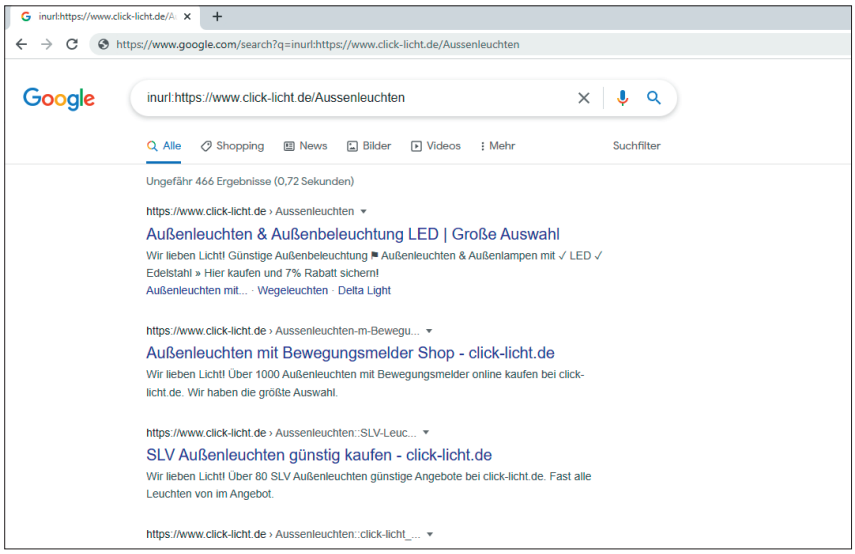

Abb. 5: Google spricht von „ungefähr 466 Ergebnissen“ bei der Kontrolle des Indexierungsstatus

Grundsätzlich können diese von Google indexiert werden. Ist dies der Fall, dann sind sie in der Google Search Console innerhalb der Indexabdeckung unter „Gültige Seite(n) mit Warnungen“ unter „Indexiert, obwohl durch robots.txt-Datei blockiert“ zu finden (*http://einfach.st/gwebm453*).

Doch das ist nicht der einzige Status, den blockierte Adressen in der GSC haben können. Denn unter „Ausgeschlossen“ gibt es mit „Durch robots.txt-Datei blockiert“ (siehe *http://einfach.st/gog443*) eine weitere mögliche Eingruppierung. Die hier genannten Adressen sind nicht im Index zu finden.

Ist die (mögliche) Indexierung dieser blockierten Seiten ein Problem? Aus meiner Sicht nicht, da dadurch verhindert wird, dass sich Crawler in der Facettierung austoben und so mindestens weitere 540.000 URLs finden. Bei Click-Licht wurde also mit wenig Aufwand viel erreicht.

Doch gibt es noch andere Möglichkeiten? Ja, die gibt es, und das Stichwort ist hier vor allem URL-Maskierung. Mit diesem Vorgehen kann gesteuert werden, ob neue URLs z. B. durch Filterungen entstehen. Für Nutzer ändert sich dabei wenig: Filter können ausgewählt werden und meistens wird auch eine neue Adresse aufgerufen, doch für Suchmaschinen ist diese Adresse nicht sichtbar, da keine normale Verlinkung (mittels <a href=“”>) zum Einsatz kommt. Das ist grundsätzlich die (noch) bessere Variante: Denn warum Suchmaschinen Adressen zeigen, die sie nachher nicht crawlen oder indexieren sollen? Man zeigt ja auch keinem Kind einen süßen Hundewelpen mit „hier, ein Geschenk“ und nimmt den Welpen dann wieder weg. Bevor jetzt Aktionismus in den SEO-Teams ausbricht: Solche Lösungen sind nur für sehr große Seiten von Relevanz!

Doch wie steht es um die Indexierung in der Außenleuchten-Kategorie? Laut einer site:-Abfrage bei Google sind für das URL-Muster 466 Adressen und damit (deutlich) mehr als die 123 eigentlich zur Indexierung freigegebenen Adressen bekannt. Allerdings lassen sich nur knapp 90 Adressen überhaupt anzeigen – wie viele Seiten genau indexiert wurden, weiß nur der Seitenbetreiber bzw. die Google Search Console.

Was lässt sich festhalten? Besonders bei Websites, die viele URLs über z. B. Filterungen generieren und nach einer schnellen Lösung suchen, kann ein Disallow die am schnellsten umsetzbare Variante sein. Und das gilt aus meiner Sicht auch dann, wenn (Filter-)Seiten via Noindex ausgeschlossen oder per Canonical zusammengefasst werden. Denn ich bin der Meinung, dass SEOs eher sparsam mit den Crawling- sowie den Indexressourcen von Google umgehen sollten.

Wichtig ist beim Einsatz der robots.txt, dass vorab genau geschaut wird, ob die Disallow-Regeln nicht Probleme an anderer Stelle verursachen. Besonders kritisch können hier .css- oder .js-Dateien sein, die versehentlich geblockt werden. Ein Allow auf die entsprechenden Dateiendungen ist eine dringende Empfehlung.

Ein Tipp: Browser-Erweiterungen wie der „Robots Exclusion Checker“ zeigen auch Blockierungen durch die robots.txt an. Das Plug-in findest du neben weiteren Empfehlungen unter *http://einfach.st/stefanczysch*.

Sofern möglich, sollten Verlinkungen auf für Suchmaschinen nicht relevanten Adresse von vornherein unterbunden werden, anstatt Adressen per robots.txt zu sperren, per Noindex von der Indexierung auszuschließen oder mittels Canonical-Tag zusammenzuführen. Zumindest dann, wenn es um eine große Anzahl an Adressen geht.

Eine unter technischer SEO-Sicht perfekte Seite kommt ohne diese Instrumente aus – doch eine solche Seite habe ich bei umfangreichen Webauftritten noch nie gesehen. Und werde es vermutlich auch nie. Auch hier nochmals der Hinweis: Diese Themen sind besonders für große Seiten mit mindestens 1.000.000 URLs relevant. ¶