

# SMX

## SUMMER EDITION – FULL POWER!



Mario Fischer

Anfang September startete in München die erste der großen Branchenkonferenzen als sogenannte Hybrid-Konferenz, die Search Marketing Expo, kurz SMX. Der eigentliche Termin im März, also die „Vor-Ort“-Konferenz, musste ja bekanntlich wegen des Lockdowns nur wenige Tage (!) vorher abgesagt werden. Statt der sonst üblichen 2.000 Teilnehmerinnen und Teilnehmer waren 350 die beiden Tage vor Ort in Präsenz, alle anderen nahmen live an den Übertragungen teil.

Hatte man ein mulmiges Gefühl in Zeiten von Corona? Nein, ganz im Gegenteil. Die Veranstalterin Sandra Finlay wartete mit einem durchdachten und umfassenden Sicherheitskonzept auf. Die nötigen Abstände konnte man zu jeder Zeit einhalten, sowohl in den Sälen als auch außerhalb. Und alle Anwesenden zogen hervorragend mit. Selbst an den Bistrotischen konnte man sich via QR-Code einloggen, damit später bei Bedarf ggf. nachvollzogen werden konnte, wer mit wem engeren Kontakt hatte.

So macht Konferenz auch in diesen schwierigen Zeiten Spaß. Und ganz ehrlich: Es tat wirklich sehr gut, sich endlich wieder mit Fachkolleginnen und -kollegen auszutauschen und sich entspannt Neues präsentieren zu lassen. Website Boosting war für Sie vor Ort und wie immer finden Sie eine höchst subjektive Zusammenfassung auf kleinstem Raum mit hoffentlich für Sie vielen Tipps und Anwendungsbeispielen.

### Speed, Speed, Speed

Tom Anthony beherrscht das einfache Erklären komplexer Sachverhalte aus dem Effeff. Und mit dieser unnachahmlichen Art erklärte er auch den nicht technikaffinen Anwesenden, wie das Web im Hintergrund funktioniert und wo die Schwachstellen beim Transport von Objekten vom Browser zu einem Webserver und zurück liegen. Beim Laden einer einzelnen Webseite wird ja nicht nur die HTML-Datei übertragen, sondern auch alle nötigen Ressourcen werden dazu mit angefordert, wie z. B. Bilder, CSS-, JavaScript-Dateien, Schriftarten und anderes. Das kann sich schnell zu 100 oder 150 Dateien ausweiten, die für den korrekten Aufbau bzw. die Darstellung einer einzelnen Webseite benötigt werden und nachgeladen werden müssen.

Er verglich das Abrufen der Datenpakete mit dem Transport von Waren via Lkw. Diese Lkw bewegen sich mit Lichtgeschwindigkeit, können aber nicht schneller fahren. Ruft man von München aus z. B. eine Webseite im Silikon Valley auf, passiert im Hintergrund Folgendes:

- » 50 ms braucht der Lkw bis zum Server, das ist die „Request Time“.
- » Der Server braucht 50 ms, um die Seite zu produzieren bzw. zur Verfügung zu stellen. Das ist die „Server Time“.
- » Der Lkw fährt in 50 ms zurück zum Browser in München, das ist die „Response Time“. Ist die Ladung umfangreicher (also eine größere Datei), kann der Transport auch schon mal 150 ms in Anspruch nehmen.
- » Wenn der Lkw zurück beim Browser ist, erfährt er, dass noch sehr viel mehr Pakete aus dem Silikon Valley geholt werden müssen. Jetzt werden mehrere Lkw losgeschickt, wozu allerdings weitere „Straßen“ in Form von Verbindungen eröffnet werden müssen. Diese müssen erst „gebaut“ werden. Das nimmt erneut etwas Zeit in Anspruch. Damit niemand in die Lkw sehen kann (Thema Sicherheit), müssen Tunnel gebaut werden (HTTPS, also Verschlüsselung). Das Bauen von Tunneln ist aber noch aufwendiger als normaler Straßenbau. Man muss dreimal hin- und herfahren

SMX  
Summer Edition



Abb. 1: Tom Anthony erklärte Sitespeed verständlich für jeden

(round trip), um einen solchen Tunnel zu bauen. Bevor die erste „sichere“ Fahrt (Request) gemacht werden kann, vergehen also 300 ms. Wenn Hunderte solcher Requests für eine Webseite gemacht werden müssen, summiert sich das sehr schnell heftig auf.

Dazu kommt, dass oft Ressourcen von anderen Webservern, nicht selten zehn, zwanzig oder mehr geladen werden. Zu denen müssen ebenfalls erst einmal einzelne Tunnel gebaut werden!

Plötzlich ist „Lichtgeschwindigkeit“ gar nicht mehr so schnell, weil durch unbedachtes Webdesign bzw. Programmierung sich die einzelnen Transporte behindern können, weil Pakete auf eine nur begrenzte Anzahl an Lkw (typischerweise sechs Stück, also sechs gleichzeitige Verbindungen) warten müssen. So werden aus Millisekunden schnell Sekunden, die man durchaus bemerkt.

Abbildung 2 zeigt einen solchen Ablauf in einer sogenannten Wasser-

fall-Darstellung. Im ersten Schritt (erste Zeile) wird ein Tunnel gebaut, was an dem orange-lila Balken erkennbar ist. Dann wird die HTML-Seite übertragen. Anschließend wird über den gleichen Tunnel das CSS-File geholt. Jetzt „vermutet“ der Browser, dass da noch mehr kommt, und baut einen neuen Tunnel für einen zweiten Lkw (Zeile 3, orange-lila Balken). Ab jetzt wird mit zwei Lkw

„Size is booming“;  
Tom Anthony

durch zwei Tunnel gleichzeitig gefahren. Ab Zeile acht ist zu sehen, dass der Browser erneut aufstockt und vier weitere Tunnel anlegt. Nun kann in sechs Tunneln gleichzeitig gefahren werden. Im Wasserfall-Modell ist sehr schön zu sehen, wie jeweils sechs Pakete transportiert werden, und wenn diese ausgeliefert sind, wird das nächste Paket auf die Reise geschickt. Aber eben erst dann ...

Was steuerbar ist, so Anthony, ist die Entfernung, die Anzahl der Tunnel (Zeit), die Größe der Pakete und deren Anzahl.

**Entfernung:** Durch den Einsatz eines CDN (Content Delivery Network) wie Cloudflare, Fastly oder andere verkürzt man die Entfernung teils dramatisch. Es werden einfach Kopien der Ressourcen angelegt, die näher beim

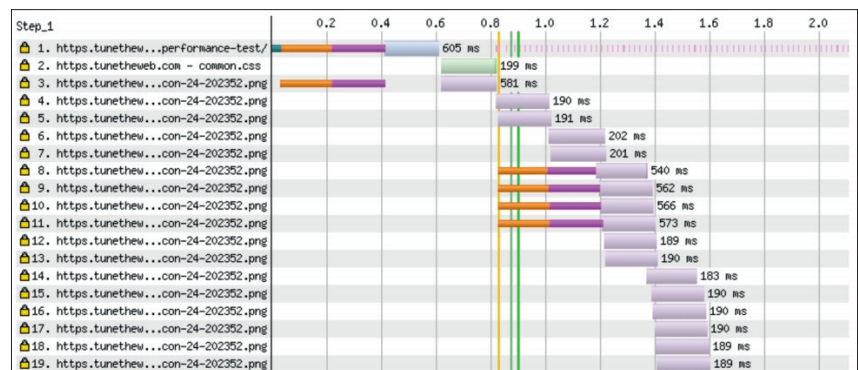


Abb. 2: Mit dem HTTP-1.1-Protokoll laufen hier sechs Anfragen gleichzeitig

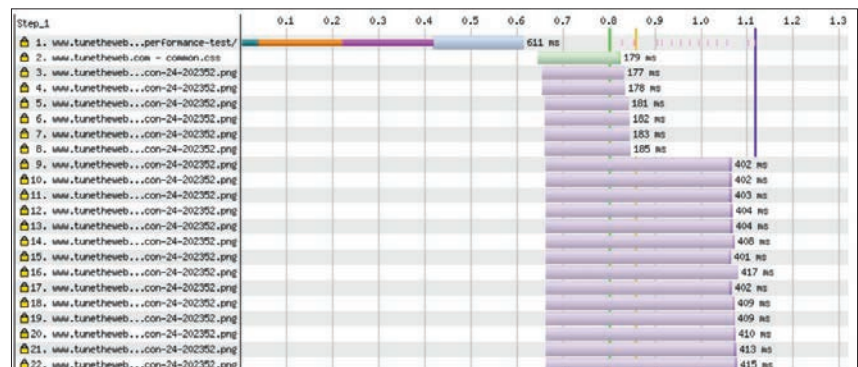


Abb. 3: Bei HTTP/2 werden Anfragen deutlich anders behandelt

„150 ms von München zum Silicon Valley – das klingt wenig“; Tom Anthony



Abb. 4: Vorsicht bei der Interpretation der CTR-Werte in der Search Console (Quelle: Jens Fauldrath)

abfragenden Browser liegen. Ein Lkw muss also nicht bis ins Silikon Valley fahren, sondern z. B. nur bis Frankfurt, weil dort eine Kopie des Paketes liegt. Mit anderen Worten können aus 50 ms schnell 20 ms werden, nicht selten aus 200 ms eben auch nur noch 20 ms. Und das gilt für alle nachfolgenden Pakete. Ein großer Irrtum ist, dass mehr Bandbreite das Problem bei großen Paketen lösen würde. Eine Datei, die bei 1 Mbps 3.000 ms benötigt, braucht bei 3 Mbps zwar nur noch 1.600 ms, aber dann schmilzt der Geschwindigkeitsvorteil schnell dahin. Bei 10 Mbps (mehr als dreimal schneller!) landet man, wie der Referent zeigte, nur bei 1.400 ms. Der Grund ist einfach: Die Lkw können noch immer nur mit Lichtgeschwindigkeit fahren und müssen oft warten, weil zu wenige Tunnel zur Verfügung stehen. Mehr Bandbreite bringt also für die Page Load Time vergleichsweise wenig. Selbst bei 1.000 Mbps geht es am Ende nicht schneller. Für Streaming hilft die Bandbreite, aber eben nicht für Webseiten.

**Anzahl der Tunnel bzw. Zeit:**

HTTP2 als Übertragungsprotokoll erlaubt, dass mehrere Lkw einen Tunnel gleichzeitig benutzen können. Zu sehen ist die Wirkung gut in Abbildung 3. Das Laden (Request) des ersten Pakets (erste Zeile) geht hier zwar langsamer, aber danach flutschen deutlich mehr Lkw durch den errichteten Tunnel. Der Stau ist beseitigt. Hat man allerdings noch immer zu viele Abhängigkeiten

„Ohne Segmentierung hat man nur einen Datenbrei“; Jens Fauldrath

innerhalb der Ressourcen, vermindert sich der Vorteil schnell wieder. **Größe der Pakete:** Wie man die oft unnötig aufgeblähte Größe von JavaScript-, CSS- und Bilddateien teils dramatisch einschrumpfen kann, ist hinlänglich bekannt. Und es gibt unzählige kostenfreie Tools, diesen Job zu erledigen. Gemacht wird es oft trotzdem nicht.

**Anzahl der Pakete:** Die gängige Meinung ist, dass kleine Dateien mit wenigen KB Volumen nicht so viel ausmachen. Wegen der Besonderheiten des Transports hat das jedoch trotzdem meist große Auswirkungen. Auch ganz kleine Pakete müssen jeweils in einen Lkw geladen werden und zwanzig Pakete brauchen zwanzig Lkw – auch wenn sie noch so klein sind. Hier liegt das Kern- bzw. Verständnisproblem vieler Webverantwortlicher. Eine Lösung kann z. B. die Verwendung von sog. Sprite Sets sein. Allerdings ist das Zusammensetzen auf der Webseite komplizierter.

Wenn man das grundsätzliche Problem des Transports von Daten im Internet wirklich verstanden hat, denkt man sicherlich etwas anders über den Auf-

bau seiner Webseiten und die Abhängigkeiten der Dateien untereinander. Damit ist der Boden für die nötigen Veränderungen bereitet.

**Die Google Search Console – einfach und doch kompliziert?**

Die Search Console von Google ist mittlerweile fester Bestandteil des Informationspools von Webmastern und SEOs geworden. Die Suchmaschine zeigt hier für die eigene Domain an, ob und welche Fehler beim Verarbeiten der Inhalte der Website aufgetreten sind. Zudem gibt es einige wichtige Möglichkeiten, hier Daten zu hinterlegen, damit Google die Strukturen besser „verstehen“ kann, oder neue URLs in den Index aufnehmen zu lassen.

Jens Fauldrath brachte es gleich zu Anfang auf den Punkt. Die Bedienung sieht einfach aus, ist es aber in der Regel nicht. Das fängt, so Fauldrath, schon bei der Einrichtung des Tools an. Da Abfragen über die Weboberfläche des Tools auf jeweils 1.000 Datensätze limitiert sind, kann sich für größere Domains ein völlig verzerrtes Bild ergeben. Hier sollte man wichtige Verzeichnisse gesondert verifizieren und alle Hosts (z. B. mit und ohne www. und andere Subdomains) und Protokolle (HTTP bzw. HTTPS) als eigene Property einrichten. Ansonsten können wichtige Informationen verloren gehen bzw. verborgen bleiben. Ein Test von Fauldrath hat bei einer großen Domain mit 10



Abb. 5: Markus Hövener erklärte alles Wissenswerte über das Crawlen

Mio. URLs ergeben, dass gegenüber dem Anlegen nur der Root-Property und zusätzlich wichtiger Verzeichnisse für den zweiten Fall bis zu 90 % mehr Suchanfragen bzw. Keywords übermittelt werden. Ein „kleines“ Problem ergibt sich dann allerdings, worauf Fauldrath explizit hinwies: Die Daten sind dann ggf. mehrfach vorhanden, weil sie mehrfach gezählt werden. Hier muss man also erst deduplizieren, bevor man die Daten weiterverwendet.

Ein weiterer Fallstrick kann die falsche Interpretation der Klickraten (CTR) sein. Diese wird massiv vom Design der Suchergebnisseite und der Suchintention beeinflusst (Abbildung 4). Auf jeden Fall sollte man grundsätzlich immer den eigenen Domain-/Unternehmensnamen und somit den Brandtraffic filtern, da dieser in der Regel eine deutlich höhere CTR hat. Also entweder alles ohne den Brand – oder nur den Brand betrachten.

Das typische Vorgehen dabei wäre:

- » Suchtyp: Web
- » Datum: letzte 12 Monate, um saisonale Verzerrungen zu vermeiden
- » Suchanfrage: -Domain/Unternehmens-/Markenname (z. B. -Allianz oder -Samsung)
- » Seite: +/produkte/ (Beispiel für eine Segmentierung nach URL-Gruppe)
- » Land: Deutschland (als Länderfilter)

Bei Bedarf kann man auch noch „Geräte“ und „Darstellung in der Suche“ mit in den Filter einbeziehen. Warum man so vorgehen sollte, also nach Sei-

tentypen zu filtern und den Brandtraffic herauszunehmen, lieferte er auch sofort mit. Nur so lernt man nach einiger Zeit, welche CTR für welchen Seitentyp gut, normal oder schlecht ist. Unterschiede in der CTR für unterschiedliche Suchintentionen von bis zu 500 % sind völlig normal. Wer nicht segmentiert, erhält daher einen nichtssagenden Mittelwert.

## Die Kunst des Crawlens

Wer sich ernsthaft mit SEO beschäftigt, kommt irgendwann nicht mehr an der Nutzung eines sog. Crawlers vorbei. Diese gibt es als Desktopsoftware (z. B. Screaming Frog oder Sitebulb) oder als nutzbaren Service via Browser (z. B. Deepcrawl oder Audisto). Ein solcher Crawler durchkämmt die eigene oder auch fremde Websites, sammelt strukturelle Kenndaten und läuft so lange allen gefundenen Links hinterher, bis er alle Seiten erfasst hat. Das hört sich einfach an, aber wie immer steckt der Teufel im Detail, wie Markus Hövener mit vielen Beispielen und wertvollen Tipps zeigte.

### Tipp 1: Nicht immer alles laden!

Irrelevante HTML-Inhalte, PDF, Bilder, AMP-Inhalte etc. bringen für SEO-Fragestellungen in der Regel keinen Nutzen. Sie kosten Zeit beim Crawlen und erhöhen zum Teil die Anzahl der URLs erheblich! Ggf. kommt man sogar mit einer kleineren, günstigeren Toollizenz aus, wenn man intelligenter crawlte.

**Tipp 2: Es muss nicht alles gerendert werden!** „Rendern“ bedeutet, dass

ein Crawler nicht ausschließlich den Quellcode einer Seite lädt und analysiert, sondern die Seite tatsächlich auch optisch aufbaut, wie ein Browser das tut. Jetzt (erst) ist erkennbar, wo durch CSS bestimmt z. B. ein Text oder ein Bild tatsächlich steht. Dieser Prozess ist sehr ressourcenintensiv und braucht entsprechend Zeit und Speicher. Hövener empfahl, statt des Renderns der kompletten Website lieber eine Auswahl von URLs mit unterschiedlichen Templates zu rendern. Findet man Probleme, sind diese meist genereller Art und treten dann bei allen Seiten dieses Typs auf. Um das zu erkennen, reichen wenige Seiten aus. Eine gängige Methode wäre auch, besonders gut oder besonders schlecht rankende URLs testweise zu rendern. Die URLs holt man sich einfach im Report „Leistung“ aus der Google Search Console.

Ganz generell, so **Tipp 3**, sind die meisten Fehler systemischer Natur, können in Vorlagen oder Programmierungen geändert werden und wirken sich dann auf viele oder gar alle Webseiten einer Domain aus. Um solche Fehler zu finden, muss man eben nicht immer alles aufwendig durchcrawlen. Hat z. B. ein Shop sehr viele Produktseiten, die alle mehr oder weniger gleich aufgebaut sind, kann man den Crawler auch anweisen, URLs nach bestimmten Mustern auszulassen. Angenommen, die Produktseiten weisen folgendes Muster auf:

*www.meinshop.de/products/produkt-4734412*

Das Muster lautet also im Verzeichnis /products/ immer [einText]-[eineZahl]. Sperrt man beim Crawlen z. B. alle Seiten, die an der Stelle [eineZahl] eine 9 haben, spart man im Schnitt 10 % ein. Natürlich lassen sich bei Bedarf auch mehrere Zahlen sperren. Nimmt man als Ausschluss z. B. 3, 6 und 9, spart man bei einer achtstelligen Produkt-ID etwa 94 % ein und die zufällige Stichprobe ist immer noch groß genug, um strukturelle bzw. systemische Fehler

„Cool URLs don't change“;  
 Tim Berners-Lee 1998,  
 Erfinder des Internets

erkennen zu können. Kennzahlen für die gesamte Website lassen sich freilich bei solchen Teilerfassungen nicht mehr bilden, aber zur Fehlersuche ist das Teilcrawlen eine probate Methode. Abbildung 6 zeigt, wie man für das Beispiel des Shops oben über sogenannte „regular expressions“ URLs mit den Zahlen 3, 6 und 9 in den Produkt-IDs gezielt ausschließen kann. In der Jubiläumsausgabe 60 hat Stephan Czysch übrigens genau erklärt, wie man die Power von regular expressions richtig anwendet.

Neben vielen anderen Tipps wies Hövener noch darauf hin, dass man auch die verschiedenen Sichten im Auge haben muss. Möchte man eine Site aus Sicht eines Suchmaschinen-Crawlers betrachten, muss man die Anweisungen „noindex“ und das Canonical-Tag respektieren bzw. den dortigen Direktiven Folge leisten – so wie es die Robots von Google & Co. in der Regel tun. Ist aber die Sicht auf den Index, also was die Suchmaschinen ggf. in ihren Ergebnisspeichern vorhalten, interessant, ignoriert man diese Anweisung besser, um alles in die Analyse zu bekommen, was potenziell auch im Index landen könnte. Beim Screaming Frog klickt man dazu z. B. unter Spider Configuration und dort im Reiter „Advanced“ die beiden Kästchen „Respect noindex“ und „Respect Canonical“ entsprechend an oder ab (Abbildung 8). Den Unterschied zeigte Hövener in einem Beispiel: Die Anzahl der indexierbaren Inhalte lag bei 474 URLs; nimmt man noch die nicht indexierbaren dazu, waren es in Summe 570.

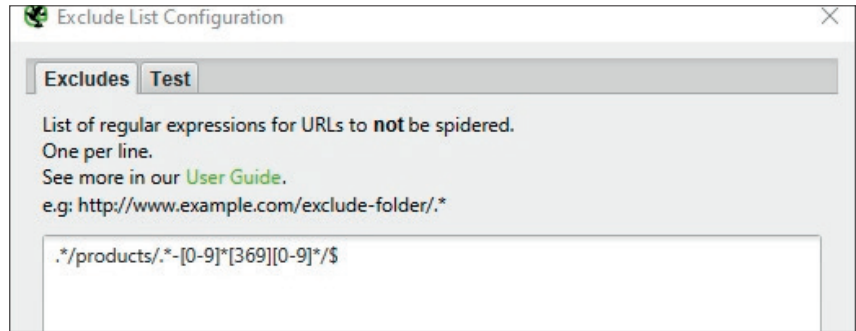


Abb. 6: URLs nach einem definierten Muster ausschließen – hier im Screaming Frog

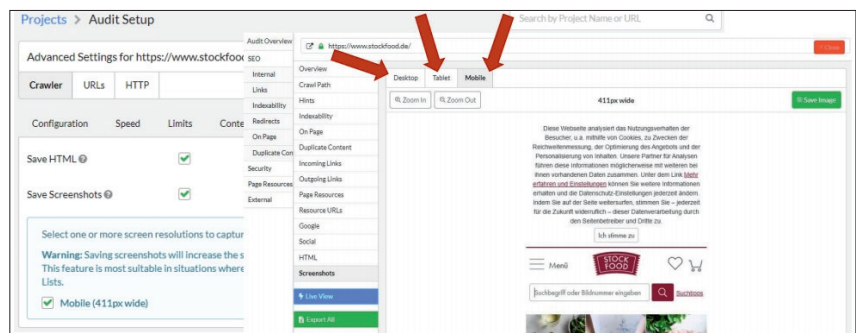


Abb. 7: Der Crawler Sitebulb erlaubt es, eine Webseite per Klick als Desktop, Tablet oder Mobile zu vergleichen (Quelle: Markus Hövener)

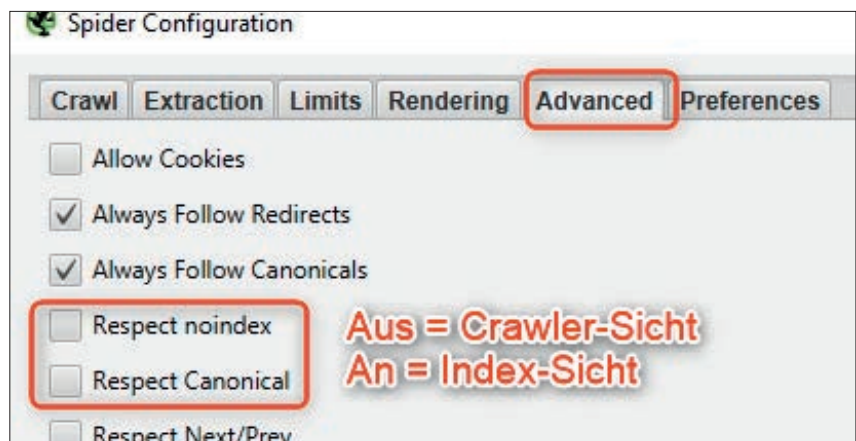


Abb. 8: Zwei Kästchen beim Screaming Frog, große Wirkung (Quelle: Markus Hövener)

### Weiterleitungsmanagement

Stefan Fischerländer hat über 20 Jahre Erfahrung im Web und ist bekannt für das Ausgraben interessanter Inhalte. Und so präsentierte er tatsächlich über den Zeitraum einer Stunde interessante Ideen und Hinweise für Optimierungen.

Es gibt viele Gründe, echte und vorgeschobene, warum man einmal veröffentlichte URLs auf andere weiterleiten muss oder möchte. Jede Weiterleitung braucht Zeit, weil der Server zunächst die neue Adresse (301-Weiterleitung) an den anfragenden Browser zurück-

schickt. Der muss dann die neue URL abfragen. Warum das u. a. problematisch ist, wurde oben in dem Vortrag vom Tom Anthony erklärt. Es fährt nicht ein Lkw, sondern zwei – und das ggf. bei mehreren Ressourcen. Ein eigens aufgesetzter Test von Fischerländer bestand darin, in seinem Blog einige Seiten weiterzuleiten bzw. dies nicht zu tun. Das Ergebnis:

- » Weiterleitung mit 301: zwei Wochen Dauer, bis die neue URL im Googleindex zu finden war.
- » Weiterleitung mit 302: ebenfalls zwei Wochen Dauer.

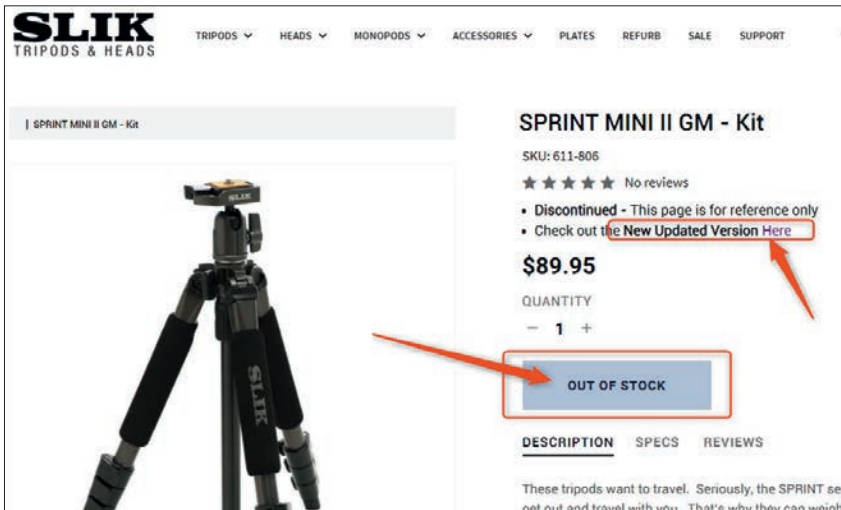


Abb. 9: Nicht immer muss eine Weiterleitung geschaltet werden (Quelle: Stefan Fischerländer)

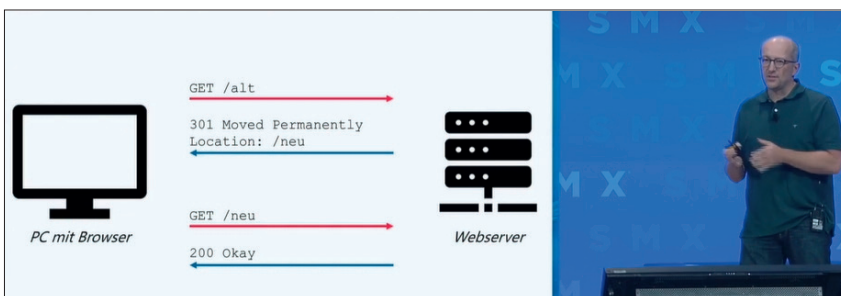


Abb. 10: Stefan Fischerländer erklärte, was bei einer 301-Weiterleitung passiert



Abb. 10: Stefan Fischerländer erklärte, was bei einer 301-Weiterleitung passiert

» Keine Weiterleitung, also einfach die URL geändert: zwei Wochen.

Bei größeren Umzügen kann das schneller gehen, meinte Fischerländer. Bei einzelnen Seiten dauert es. Laut Google ist eine 301-Weiterleitung auch keine zwingende Anweisung. Man betrachte mehrere Faktoren und nehme das nur als – wenn auch starke – Empfehlung. Es kann also sein, dass bei widrigen Umständen eine Weiterleitung sogar ignoriert wird.

Wer jetzt erst auf HTTPS umstellt, sollte berücksichtigen, dass es sich hierbei eigentlich auch um einen Relaunch

handelt. Alle URLs sind neu. In jedem Fall sollte man daran denken, in der Google Search Console eine neue Property für die neue Adresse anzulegen. Ansonsten bekommt man nach der Umstellung keine Daten mehr.

Ebenso sollte man immer auch an alle Nicht-HTML-Inhalte bzw. „versteckte“ URLs wie z. B. Bilder, hreflang-Tags, rel=alternate, Sitemaps und deren Umleitung denken.

Es muss ja auch nicht immer jedes Produkt, das nicht mehr verfügbar ist, automatisch weitergeleitet werden. Abbildung 9 zeigt, dass es auch anderes

gehen kann. Wenn ein Besucher gezielt ein bestimmtes Produkt sucht, ist es nämlich nicht immer die beste Idee, diese erste Suchabsicht zu ignorieren und ihn automatisch zu einem anderen Produkt zu „beamen“. Slik kennzeichnet via „Out of Stock“ statt eines Preises, dass dieses Stativ nicht verfügbar ist. Zusätzlich gibt man dort den Hinweis, dass es eine neue Version dieses Stativs gibt, und leitet per Link dorthin („New Updated Version Here“). Aus Usability-sicht ist das sicherlich fast nutzlos, weil der Hinweis viel zu klein und zu gut versteckt ist. Hier zählt der Wille. Wer das besser umsetzt, fährt sicherlich damit gut und spart sich mit dieser Strategie ggf. Tausende von Umleitungen.

Wie man einen Relaunch gründlich verpatzen kann, zeigte Fischerländer ebenfalls. Die Stadt Frankfurt, die beim Relaunch Weiterleitungen nicht, nicht korrekt oder auf nicht vorhandene Seiten setzte, stürzte mit ihren vormals durchaus respektablen Rankings dramatisch ab, wie in Abbildung 11 gut zu sehen ist.

### Fazit

Wenn auch mit vermindertem Besucherumfang vor Ort, die SMX kann durchaus nicht nur als voller Erfolg bezeichnet werden, es wurden auch alle gesundheitsrelevanten Regeln eingehalten. Die erste große Konferenz seit Corona zu veranstalten, erforderte wirklich extremen Mut von der Veranstalterin Sandra Finlay. Hierfür und für das gute Gelingen darf die Branche ihr dankbar sein. Alle Teilnehmerinnen und Teilnehmer zogen mit und es stellte sich durchaus schnell ein gewisses Wohlfühl- und Sicherheitsgefühl ein. Eine SMX auch in Zeiten von Corona? Jederzeit gerne wieder.

**Save the date:**

**Die nächste SMX wird am 17.–18. März 2021 in München stattfinden. Weitere Infos unter [smxmuenchen.de](https://www.smxmuenchen.de)**