



# AUTOMATISIERTES MONITORING MIT SCREAMING FROG UND GOOGLE DATA STUDIO

Nicht mal die Entwickler von Screaming Frog (SF) wissen genau, wie viele verschiedene Metriken sich mit ihrem beliebten Crawler-Tool beobachten lassen. Ein Versuch des Autors, sie zu zählen, wurde beim Überschreiten von 300 abgebrochen (mitgerechnet wurden die Metriken, die sowohl nativ als auch durch die APIs gesammelt werden). Da ist also offenbar eine ganze Menge mehr zu holen, als man gemeinhin vermutet.

Evgeniy Orlov hat für Sie recherchiert und ausprobiert,

- » wie sich die Metriken von SF und der angeschlossenen APIs automatisch monitoren lassen,
- » wie sich die Monitoringdaten in der Form von Time-Series-Charts in Google Data Studio (GDS) darstellen lassen und
- » wie man das automatische Update des Google-Data-Studio-Berichts aufgesetzt.

Wer dem Step-by-step-Beitrag aufmerksam folgt, erhält am Ende einen Google-Data-Studio-Report, der als Datenquelle regelmäßige automatisierte SF-Crawls hat! Somit kann man jederzeit auf aktuelle und individuell angereicherte Kenndaten einer Website zugreifen.

## DER AUTOR



**Evgeniy Orlov.** Team Lead SEO bei mediaworx berlin AG, berät Kunden bei strategischen Fragen der Automatisierung von SEO-Prozessen und Tool-Chain-Integration. Er ist der einzige Gastautor, der für Screaming Frog schreiben durfte.

Evgeniy Orlov

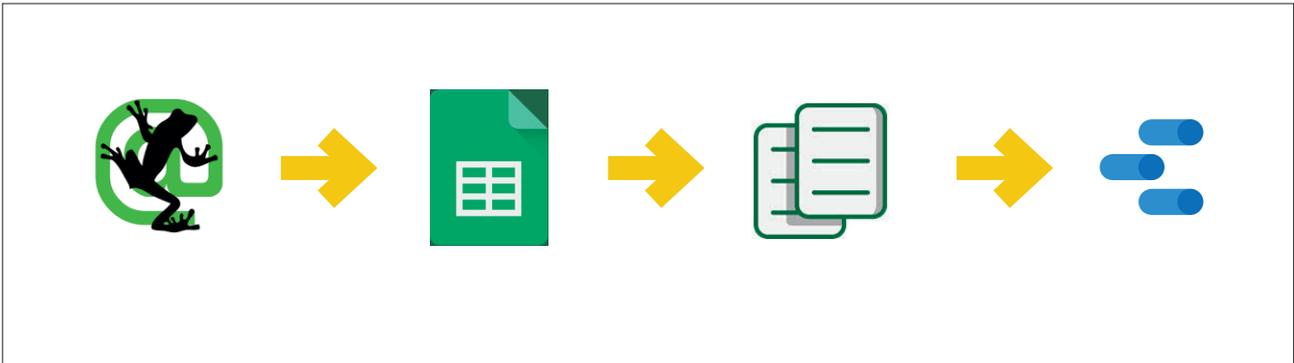


Abb. 1: Ein einfacher Datenfluss: von SF zu Google Sheets, von Google Sheets ins Archiv, aus dem Archiv in GDS

## Neue SF-Funktionalitäten erlauben jetzt nützliche Automatisierungen

Durch einige der letzten Updates von Screaming Frog (SF) wurden wichtige neue Funktionen implementiert und diese lassen sich nun bestens zur Automatisierung ansonsten aufwendiger manueller Prozesse nutzen. Die wesentlichen Erweiterungen sind:

- » Der SF kann jetzt Exporte von zeitgesteuerten automatisierten Crawls auch in Google Sheets speichern.
- » Beim Überschreiben der alten Daten mit den neueren Daten in demselben Google-Sheets-Dokument wird die ID des Dokuments beibehalten. Anfangs war diese Funktionalität gar nicht offensichtlich – erst nach eingehenden Tests stellte sich heraus, dass nur mit der Beibehaltung ein und derselben Dokument-ID tatsächlich weitere Aktionen möglich sind.
- » Die Datenansicht „internal\_all“ bekam eine Datumsspalte spendiert, wo das Datum des Crawls als „Crawl\_timestamp“ eingetragen wird. Diese Funktionalität war ebenfalls zu Anfang nicht gegeben – in einem Workaround war es nötig, zusätzlich HTTP Header zu crawlen, um das Datum des Zugriffs bestimmen zu können. Denn genau dieses Datum ist die ausschlaggebende Metrik bei der Erstellung von Time-Series-Charts in Google Data Studio (GDS)!

## Der Datenfluss im Überblick

Zum Überblick, wie die vom SF gecrawlten Daten am Ende im GDS lan-

den, wird im Folgenden zunächst kurz der Ablauf beschrieben:

- » Der SF crawlt alle notwendigen Metriken und legt sie in einem Tab eines Google-Sheets-Dokuments ab.
- » Wie bereits erwähnt, würde normalerweise der SF die Daten von heute am nächsten Tag mit den neuen Daten einfach überschreiben. Deshalb müssen die aktuellen Daten jeweils in einem anderen Tab archiviert werden. Die Überschreibung der Daten ist übrigens die einzige für die Automatisierung passende Einstellung in SF. Die zweite vorhandene Einstellung ist, dass der SF für jeden Crawl einen Ordner mit dem Crawl-Datum als Ordnernamen erstellt und dort die Datendatei ablegt.
- » In „Archiv“-Tabs sind somit die Daten jedes Crawls sauber nacheinander gespeichert.
- » Beim Erstellen des GDS-Berichts wird als Datenquelle dann der Archiv-Tab angegeben.

## Nötige Vorbereitungen für die Crawl-Einstellungen

### PageSpeed API-Schlüssel erstellen

Als Beispiel für das Berichtsmonitoring wird hier exemplarisch die Metrik Time to First Byte (weiter TTFB) genommen. TTFB, die Zeit bis zum ersten Byte, ist eine Ladezeit-Metrik, die zeigt, wie schnell der Server auf eine Anfrage reagiert. Diese Metrik erhält der SF durch die Abfrage der PageSpeed Insights (weiter PSI) API.

Dazu muss allerdings erst ein sog. API-Schlüssel für PageSpeed Insights API erstellt werden, sofern man als Entwickler darüber nicht bereits verfügt. Wie man an diesen Schlüssel kommt, beschreibt Google recht anschaulich unter <http://einfach.st/speedkey> in einer Anleitung, der man einfach folgt.

### PageSpeed API-Schlüssel in den SF eintragen

Ist der Schlüssel erstellt bzw. war schon vorhanden, muss er im SF hinterlegt werden. Unter *Configuration* -> *API Access* -> *PageSpeed Insights* findet sich ein Eingabefeld, wo man genau dies bewerkstelligen kann.

Nachdem der API-Schlüssel dort einkopiert wurde (Abbildung 3) klickt man auf den Connect-Button und der Status wechselt von Rot (not connected) zu Grün (connected), wie Abbildung 4 zeigt. Unter dem Tab „Metrics“ findet man dann ausklappbar alle verfügbaren Metriken, die Google für eine URL zur Verfügung stellt. Nach der Wahl des Endgeräts, also ob man Daten für Desktop oder Mobile möchte, klickt man dann einfach die Metriken an, die man übertragen haben möchte. Im vorliegenden Fall ist dies nur „Time to First Byte“ in der Rubrik „Lighthouse Metrics“ (Abbildung 5). Wer später mehr Metriken automatisiert über das Google Data Studio überwachen möchte, wählt sich an dieser Stelle dann die gewünschten Daten per Mausklick aus.

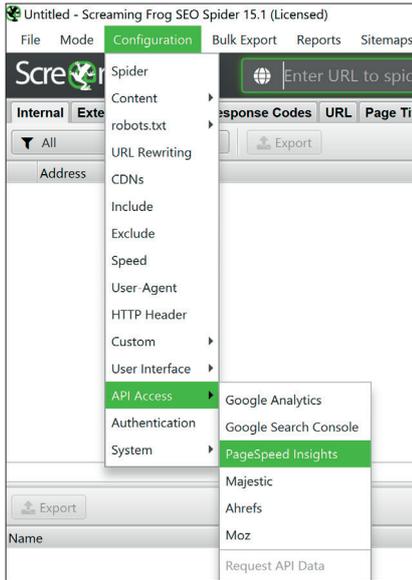


Abb. 2: Hier wird der PageSpeed-Key im SF hinterlegt

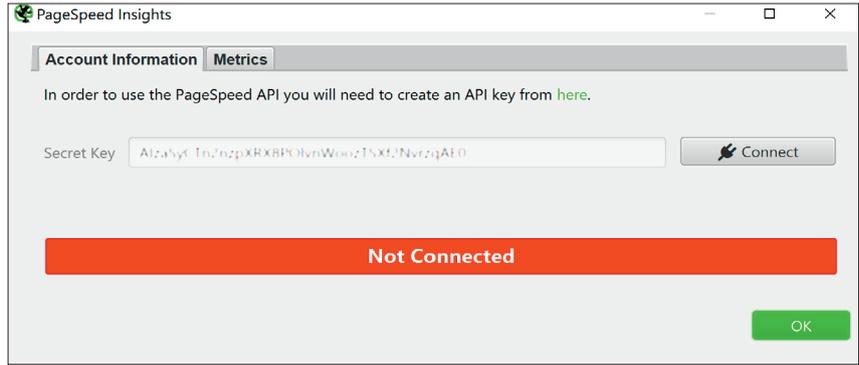


Abb. 3: Hier wird der API-Schlüssel einkopiert

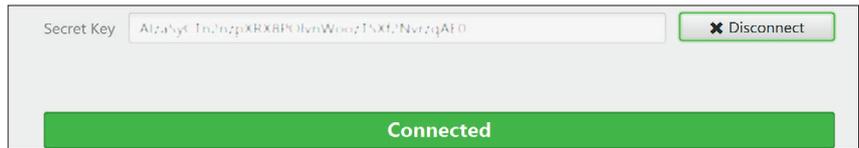


Abb. 4: Wenn der API-Schlüssel hinterlegt wurde, kann der SF mit Google verbunden werden

### Eine Konfiguration für automatische Crawls erstellen

Es ist immer ratsam, für automatisierte Crawls eine eigene Konfigurationsdatei anzulegen. Für das vorliegende Beispiel wird der Einfachheit halber tatsächlich nur die Lighthouse Metric „Time to First Byte“ aktiviert, alle anderen Checkboxes müssen deaktiviert werden.

Danach wird genau diese Konfiguration im SF-Menü unter dem Pfad *File -> Configuration -> Save as* als Datei gespeichert.

Die Konfigurationsdatei hat als Dateiendung *.seospiderconfig* – das ist zwar nicht gerade kurz, aber dafür gut erkennbar.

### EIN HINWEIS FÜR NUTZER VON NOTEBOOKS

Crawls werden nicht starten, solange das Notebook nicht mit der Stromleitung verbunden ist. Der Screaming Frog klinkt sich in das „Task Scheduling“ von Windows ein und so, wie ein Smartphone ein Update verweigert, wenn der Akku nicht ausreichend geladen ist, so ist auch hier der Stromanschluss eine eingebaute Sicherheitsmaßnahme, dass ein Crawl korrekt ausgeführt wird und nicht wegen eines schwachen Akkus abbricht.

### Crawl Mode: Als Liste oder als Crawler?

Um die bestmögliche Vergleichbarkeit und Darstellung zu erreichen, eignet sich am besten der Crawl Mode „List“. Hier lassen sich immer gleiche definierte URLs hinterlegen, die dann abgearbeitet werden. Hierzu wählt man am besten typische/wichtige Seiten der Website aus. Bei der Metrik TTFFB macht das zwar keinen Unterschied, aber wenn man ggf. später auch Metriken holt, welche z. B. vom Aufbau der Seite beeinflusst werden, bekommt man für jede Kategorie vergleichende Werte. Es macht meist wenig Sinn, z. B. bei einem Shop Daten für immer gleich aussehende 10.000 Produktseiten zu holen. Die werden sich nicht nennenswert unterscheiden, daher reichen einige URLs zum Messen aus dieser Kategorie aus. Weitere wichtige Seitentypen bei Shops wären z. B. Kategorieseiten, redaktionelle Inhalte, die Startseite, Übersichtsseiten oder Suchseiten.

Als abzuarbeitende Liste dient eine einfache Textdatei mit einer URL je Zeile, eine Excel-Tabelle oder auch eine Sitemap. Diese Liste stellt man vor dem Crawl bzw. dem Start des SF zusammen.

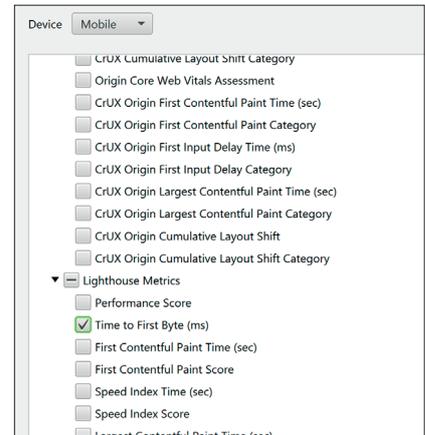


Abb. 5: Anschließend muss man noch das Gerät (Desktop, Mobile) und Metriken auswählen, die von der API pro URL gezogen werden sollen

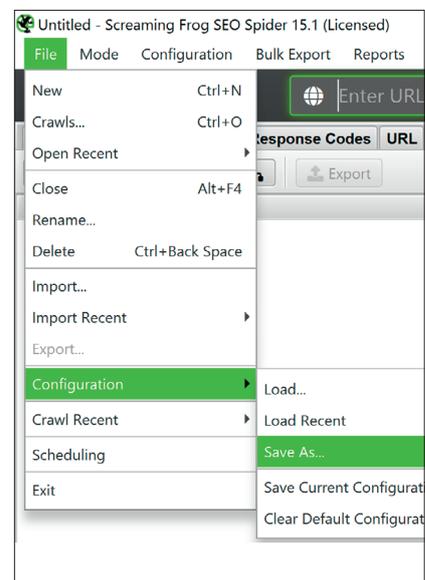


Abb. 6: Der Pfad zur Speicherung der Konfigurationsdatei

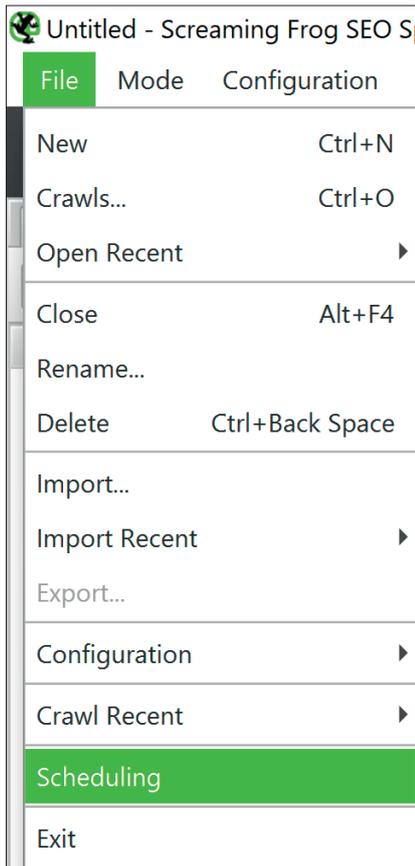


Abb. 7: Hier werden die Einstellungen für automatische Crawls verwaltet

## Den SF für den automatischen Crawl vorbereiten

Beim Screaming Frog können regelmäßige (scheduled) Crawls unter dem Menü *File* -> *Scheduling* erstellt werden.

Sofern bisher noch keine automatischen Crawls gespeichert wurden, ist hier noch kein Eintrag vorhanden. Ein Klick auf „Add“ öffnet ein weiteres Fenster mit drei Tabs, in denen alle nötigen Einstellungen vorgenommen werden können.

### Der Tab „General“

Hier werden Task-Name und Project-Name eingetragen (Abbildung 8): Beide Felder bzw. Namen sind wichtig, damit die gecrawlten Daten später am korrekten Ort in Google Drive landen. Beim erstmaligen Speichern der Daten in Google Drive erstellt SF von sich aus im Root einen Ordner namens „Screaming Frog SEO Spider“. In diesem Ordner werden Unterordner für

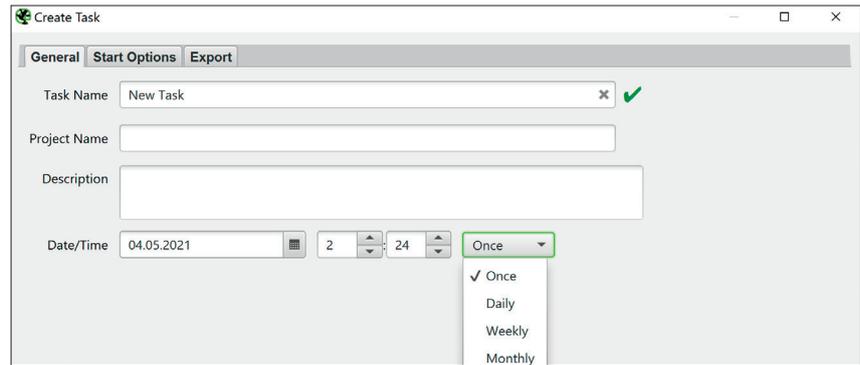


Abb. 8: Scheduling Set-up, Tab „General“

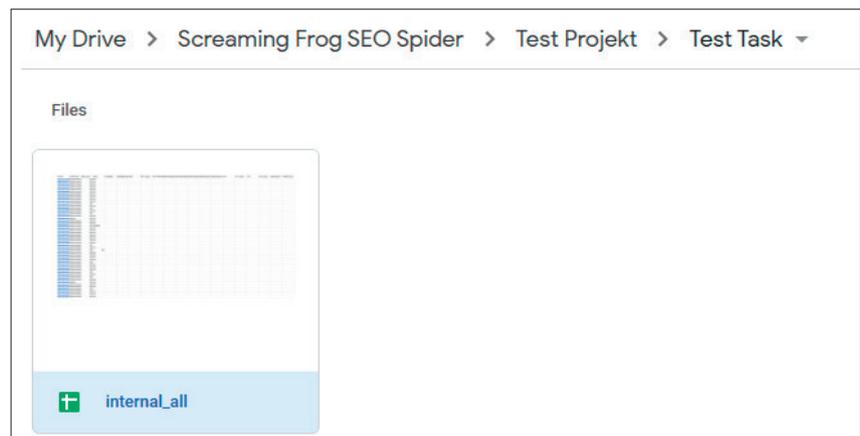


Abb. 9: GDrive-Ordnerstruktur, internal\_all ist die exportierte Datendatei mit den Ergebnissen des Crawls

Projekte erstellt, darin Unterordner für die Tasks und darin die exportierten Datendateien.

Bei Benennung von Projekt und Task als „Test Projekt“ und „Test Task“ wird in Google Drive die unter Abbildung 9 ersichtliche Ordnerstruktur erstellt.

Das Datum und die Uhrzeit in der letzten Zeile bestimmen, wann der erste Crawl starten soll. In der Drop-down-Liste wird ausgewählt, ob der Crawl einmalig, täglich, wöchentlich oder monatlich durchzulaufen hat. Um schnell die Ergebnisse zu sehen und um ggf. detailliert zu dokumentieren, wurde in diesem Beispiel „Daily“ ausgewählt.

Im Tab „Start Options“ werden anschließend die gewünschten Einstellungen für den automatischen Lauf des SF hinterlegt. Die folgenden Einstellungen wurden hier für das vorliegende Beispiel gewählt:

**Crawler Mode:** Wie erwähnt, ist es besser, mit einer Liste zu arbeiten.

Daher ist die Einstellung hier „List“.

**Crawl Seed:** Hier wird mithilfe des Buttons „Browse“ die Datei mit der zu crawlenden URL-Liste ausgewählt. Dies kann wie bereits erklärt eine einfache Text-Datei sein, mit einer URL pro Zeile und getrennt mit einem einfachen Zeilenumbruch, eine Excel-Datei oder eine XML-Sitemap.

**Crawl Config:** Hier wählt man die vorhin abgespeicherte Konfigurationsdatei aus.

Weil der SF beim Crawl hier auf die API von Google zugreifen wird, muss auch die Checkbox „PageSpeed Insights“ ausgewählt werden.

### Der Tab „Export“

Beim Klick auf diesen Tab sind zunächst fast alle Felder ausgeblendet (Abbildung 11).

Ein automatisierter Crawl startet beim SF immer in einem sogenannten headless-Modus – in einem eigenen, quasi unsichtbaren Prozess ohne eine

grafische Benutzeroberfläche. Das ist deshalb so, damit automatisierte Crawls einer SF-Instanz, die gerade in einem „normalen“ Nutzermodus ausgeführt wird, nicht in die Quere kommen. Der SF kann also im Hintergrund automatisch crawlen, während man ihn via Oberfläche manuell benutzt.

Nach dem Anklicken der headless-Checkbox werden dann auch die weiteren Optionen aktiv auswählbar (Abbildung 12).

„Output Folder“ ist hier irrelevant, weil der Output direkt in Google Drive gespeichert wird. Dafür wurde ja der Projekt- und Taskordner dort angelegt.

Die Radioboxen darunter sind sehr wichtig: Hier wird eingestellt, was mit den Daten passiert, die aus den regelmäßigen Crawls hinzukommen. Hier muss **unbedingt** die Option „Overwrite files in output“ ausgewählt werden.

Weiter unten findet man eine weitere wichtige Einstellung, mit der festgelegt wird, welche der unterschiedlichen Datenansichten aus dem SF exportiert wird. Beim Klick auf den Button „Export tabs“ öffnet sich ein Pop-up-Fenster. Im linken Teil dieses Fensters sind dann alle im SF vorhandenen Datenansichten aufgelistet. Hier wählt man die erste Datenansicht „Internal:All“ aus (Abbildung 13).

Beim Klick auf den nach rechts zeigenden Pfeil verschiebt sich die ausgewählte Datenansicht auf die rechte Fensterseite.

Danach kann das Pop-up-Fenster wieder geschlossen werden.

Unter Format findet man weiter rechts eine Dropdown-Liste für die Auswahl eines Ausgabeformats für die Exportdatei. Hier wählt man sinnigerweise „gsheet“ aus. Anschließend muss man dem SF natürlich erlauben, Daten auf dem Google Drive zu speichern. Beim Klick auf „Manage“ wird daher ein Fenster geöffnet, wo man aus den bereits vorhandenen bzw. bereits benutzten (eigenen) Google-Konten

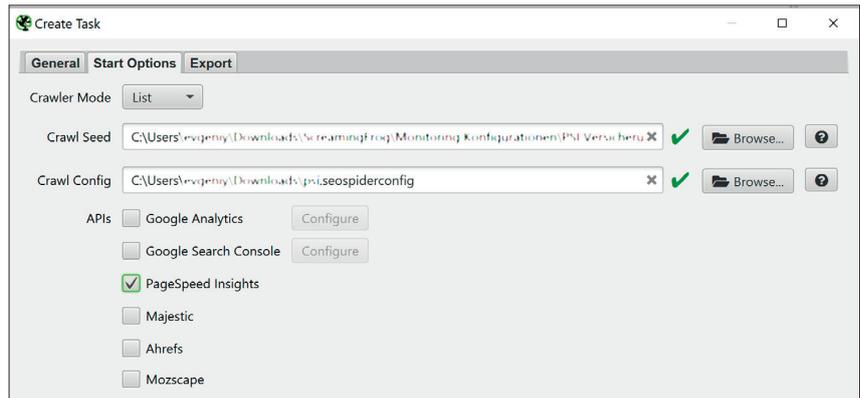


Abb. 10: Die Startoptionen im Überblick

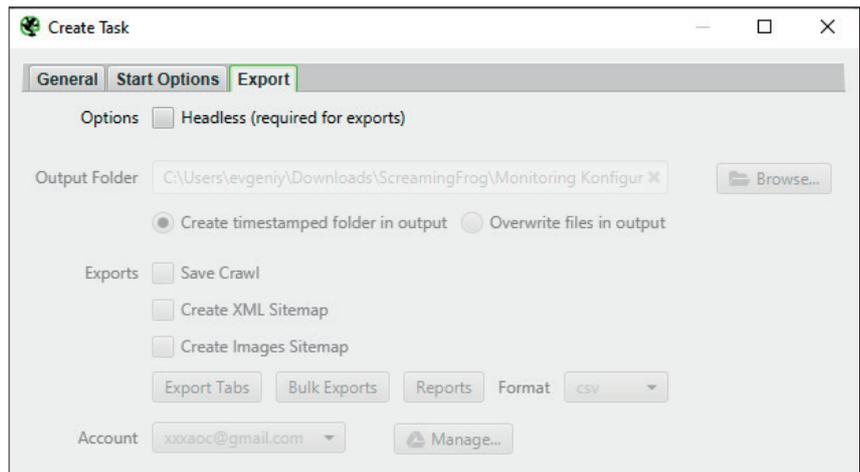


Abb. 11: Tab „Export“ – alle Optionen bis auf headless sind deaktiviert

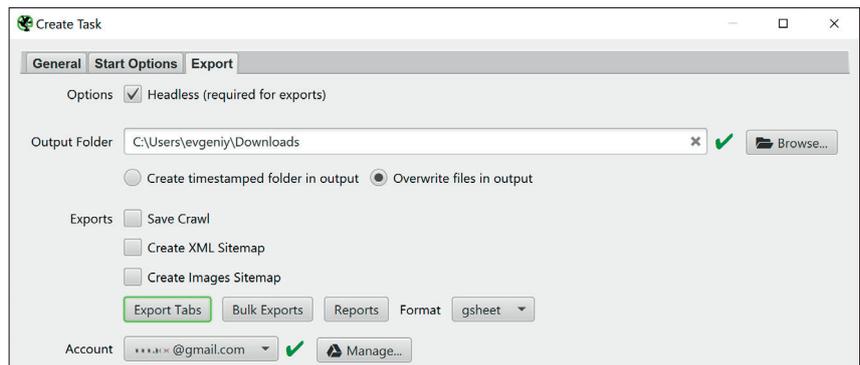


Abb. 12: Tab „Export“ – jetzt sind alle Optionen aktiv

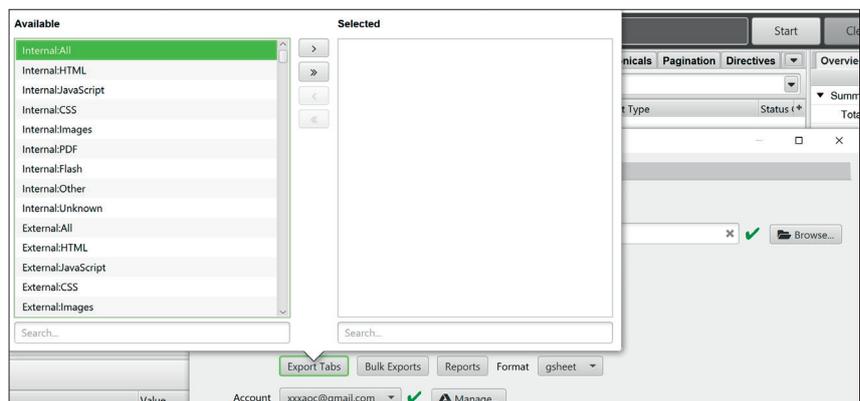


Abb. 13: Wichtig ist die Auswahl der Datensicht „Internal:All“

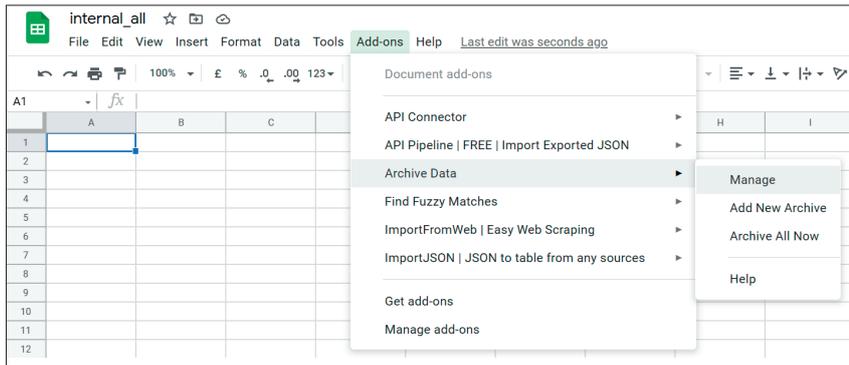


Abb. 14: Der Klick-Pfad zu den Archiv-Einstellungen Google Sheets

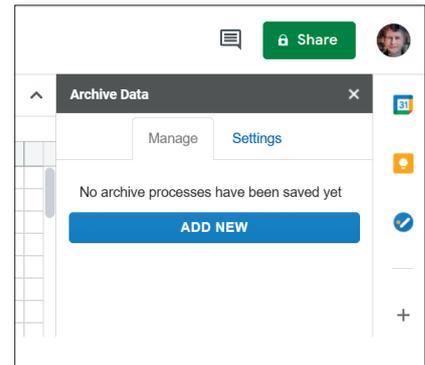


Abb. 15: Die Option „ADD NEW“ erscheint

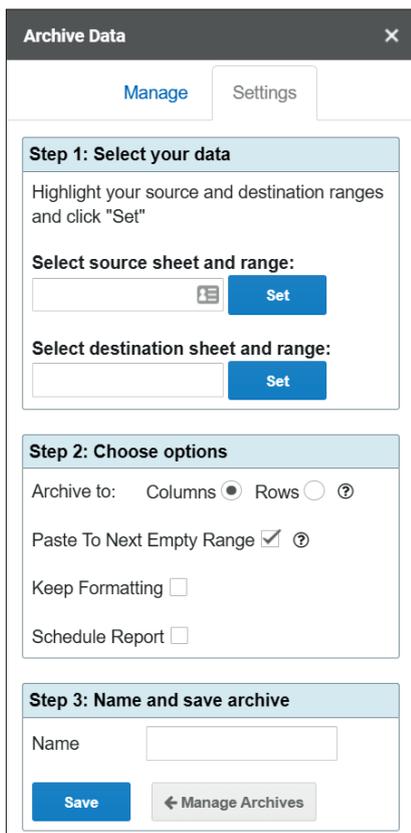


Abb. 16: Alle Archiv-Einstellungen – vorher

eines auswählen oder auch ein neues hinzufügen kann. SF leitet so transparent durch die Prozedur, dass bei jedem Schritt Klarheit über das nötige Vorgehen vorhanden ist.

Jetzt ist das Set-up des automatischen Crawls abgeschlossen – ein Klick auf „OK“ und der Projektname erscheint im Fenster mit der Liste aller automatischen Crawls.

Nach getaner Arbeit muss im Google Drive im entsprechenden Ordner eine neue Datei namens „internal\_all“ erscheinen. Diese Datei muss, um später eine passende Datenquelle für das

Data Studio zu werden, noch einige Anpassungen erfahren. Darum geht es im Folgenden.

## Google Sheets: Erstellung einer Datenquelle für GDS

In Google Drive liegt nun ein Google-Sheets-Dokument, das sich nach jedem Crawl aktualisiert bzw. ergänzt. Da bewusst nach einer URL-Liste gecrawlt wird, bleibt die Anzahl der Zeilen und Spalten im Sheet 1 immer gleich.

Damit eine GDS-Datenquelle sich für die Erstellung von „Time-Series“-Charts eignet, müssen im vorliegenden Kontext zwei Bedingungen erfüllt sein:

1. Jede Datenzeile muss ein Datum haben.
2. Die Crawl-Daten müssen so archiviert werden, dass sie in einem anderen Tab immer untereinander geschrieben werden, nachdem die neuen Daten nach dem jeweils nächsten Crawl erscheinen.

Die Bedingung 1 ist erfüllt – wie erwähnt, enthält seit Neuestem die SF-Datenansicht internal\_all in jeder Zeile das Datum und die Uhrzeit des Crawls (Crawl\_timestamp), und zwar praktischerweise gleich in einem von Google Sheets erkennbaren Format.

Um die Bedingung 2 zu erfüllen, sind einige vorbereitende Modifikationen nötig.

### Archive Data: Add-on finden und installieren

Ana Kravitz (<https://mixedanalytics.com/>) entwickelte ein Google Sheets

Add-on „Archive Data“ (ladbar unter <http://einfach.st/anakravitz>), das ziemlich genau das gesetzte Ziel erfüllt. Das Add-on muss installiert werden – es kostet nichts.

### Archivierungseinstellungen

Nach der Installation des Add-ons von Ana Kravitz wechselt man in die Datenansicht. Der Inhalt der Datei sieht ungefähr so aus:

- » Spalte A: „Address“, es sind die gecrawelten URLs
  - » Spalte B: Content Type
  - » Spalte C: Status Code
  - » Spalte D: Status
- Dann folgen viele Spalten mit den Spaltenüberschriften von SF, aber ohne Inhalte, und weiter rechts, als letzte drei Spalten, folgen dann:
- » Time to First Byte mit den gecrawelten Werten
  - » URL Encoded Address – noch einmal die URLs, und, als letzte Spalte
  - » Crawl Timestamp mit dem Datum und der Uhrzeit.

### Hinweis: Alle folgenden Schritte müssen in der exakt beschriebenen Reihenfolge ausgeführt werden:

- » Einen neuen Tab erstellen
- » Den neuen Tab als „Archiv1“ umbenennen
- » Im Sheet1 die Zeile 1 mit den Spaltenüberschriften kopieren und im Archiv 1 in die Zeile 1 einsetzen
- » Archiv-Einstellungen öffnen: *Add-ons* -> *Archive Data* -> *Manage*

**DIE ZEITEN RICHTIG EINSTELLEN!**

Der Zeitraum für die Archivierung sollte erfahrungsgemäß am besten eine Stunde nach dem Crawl eingestellt werden. Wurde z. B. der Crawl auf 8 Uhr vormittags eingestellt, so sollte die Archivierung in der Zeit von 9 a. m.–10 a. m. erfolgen. Also immer erst der Crawl, dann erst die Archivierung!

Abbildung 14, Abbildung 15 und Abbildung 16 zeigen das weitere Vorgehen. Die Seitenleiste rechts mit den Archiv-Einstellungen bleibt offen, unabhängig davon, in welchem Tab man sich befindet.

**Select source sheet and range:** Sinngemäß muss hier der Tab und der Bereich ausgewählt werden, der zu archivieren ist. Im Tab „Sheet1“ wird hierfür der ganze Bereich ausgewählt, die die Daten enthält, jedoch nicht die Zeile 1 mit den Spaltenüberschriften. Wenn dieser Bereich z. B. über A2:BA70 geht, wählt man ihn aus und klickt den Button „Set geklick“ an – Sheet1!B2:BA70 bleibt in dem Feld stehen.

**Select destination sheet and range:** Dasselbe muss für den Ort der Archivierung gemacht werden – es muss im Tab „Archiv1“ der Bereich derselben Größe für das Ziel ausgewählt werden wie die Quelle. Hierfür wird der Bereich A2:BA70 ausgewählt und auf den Knopf „Set“ geklickt. Archiv1!A2:BA70 bleibt im Feld stehen.

**Archive to:** „Rows“ muss ausgewählt werden, damit die archivierten Daten später untereinander erscheinen.

**Paste To Next Empty Range:** Ist bereits voreingestellt.

**Keep Formatting:** Hier nichts anklicken.

**Schedule Report:** Wird verwendet, denn die Daten müssen ja immer wieder archiviert werden, nachdem im Sheet1 die neuen Daten die alten überschreiben.

**Run Report:** Nachdem die Check-

box gesetzt ist, erscheint eine Dropdown-Liste „Run Report“ mit der Auswahl, ob die Archivierung täglich, wöchentlich oder monatlich ausgeführt werden soll.

**Between:** Wird „täglich“ gewählt, erscheint noch ein Dropdown namens „between“, wo die Zeiträume ausgewählt werden dürfen, die jeweils eine Stunde lang sind. Wird z. B. 9 a. m – 10 a. m. ausgewählt, so erfolgt die Archivierung im Zeitraum von 9 bis 10 Uhr vormittags.

**Name:** Hier wird der Name vergeben und mit „Save“ gespeichert. Somit ist Archivierung korrekt eingestellt und abgeschlossen. Abbildung 18 zeigt alle vorgenommenen Archiv-Einstellungen im Überblick.

Der Tab „Archiv1“ ist nun korrekt modifiziert, um als Datenquelle für einen GDS-Bericht zu dienen.

**Den GDS-Report erstellen**

**Report aufsetzen**

Um den Report für Data Studio aufzusetzen, lädt man es zunächst unter <https://datastudio.google.com>. Dort erstellt man diesen neuen Report unter -> Create -> Report (Abbildung 18).

Anschließend wählt man als Connector Google Sheets aus, wie Abbildung 19 zeigt. Nun wird ein Auswahlscreen angezeigt, wo alle zur Verfügung stehenden Google Sheets aufgelistet sind. Hier wählt man das vorher angelegte Dokument „internal\_all“ aus und dort dann den Archiv-Tab „Archiv1“ (Abbildung 20).

Die Auswahl wird dann ganz einfach durch einen Klick auf den Add-Button unten rechts bestätigt. Nach dieser Bestätigung erscheint bereits der Report mit der ersten Datentabelle (Abbildung 21).

In der rechten Spalte „Available Fields“ sind alle Felder aus der Datenquelle aufgelistet – bei der Einbindung der Datenquelle wurde die Option

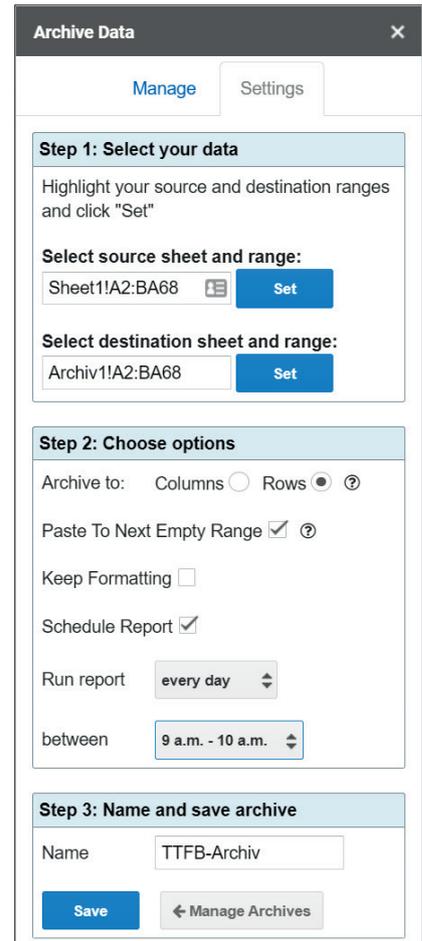


Abb. 17: Alle Archiv-Einstellungen – naher

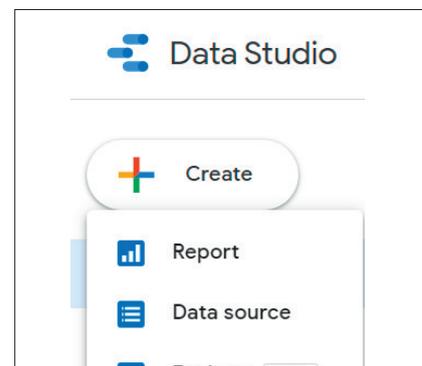


Abb. 18: GDS – einen Report erstellen

„Use first row as headers“ ausgewählt. Zwar hat die Tabelle viele Spalten mit den Überschriften, aber wenig Daten – genauer gesagt, nur die für die Darstellung nötigsten.

Entsprechend wird die Visualisierung der beobachteten Metrik zusammengesetzt: auf der X-Achse die Zeit, auf der Y-Achse die TTFB-Werte der beobachteten URLs.

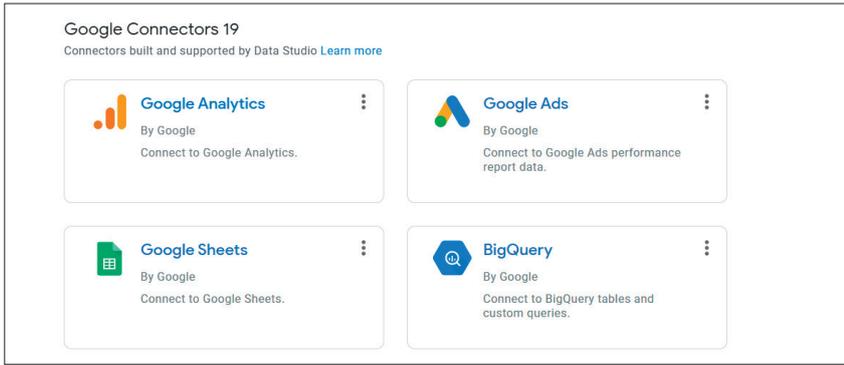


Abb. 19: GDS – Google Sheets als Connector (Datenquelle) auswählen

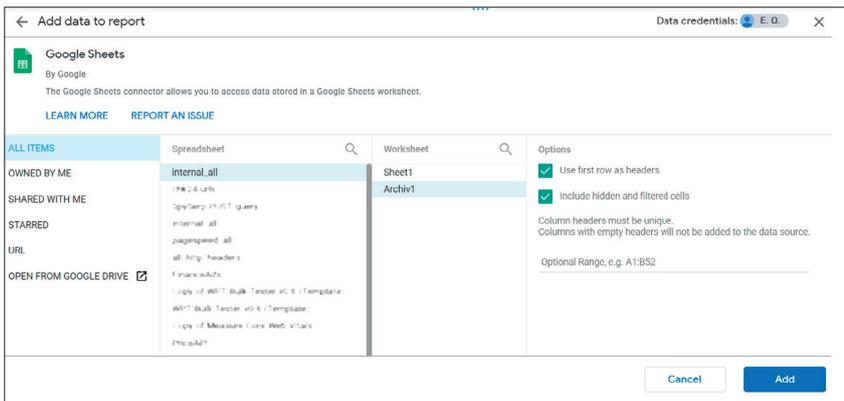


Abb. 20: GDS – im Google-Sheets-Dokument den entsprechende Archiv-Tab wählen

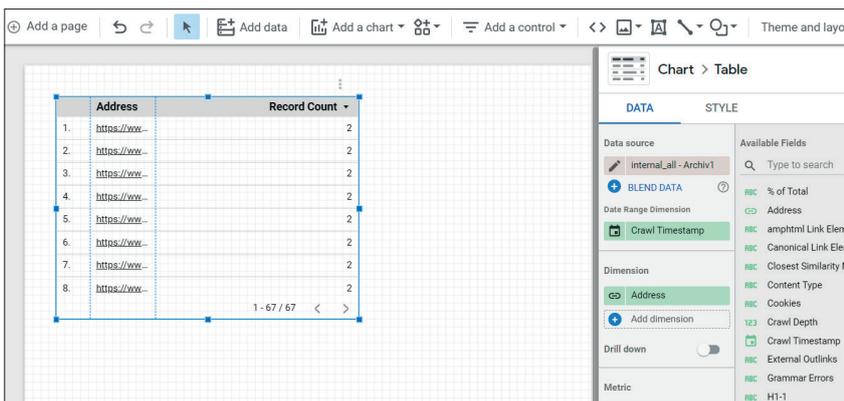


Abb. 21: GDS – die erste automatische Report-Darstellung nach Anbindung der Datenquelle

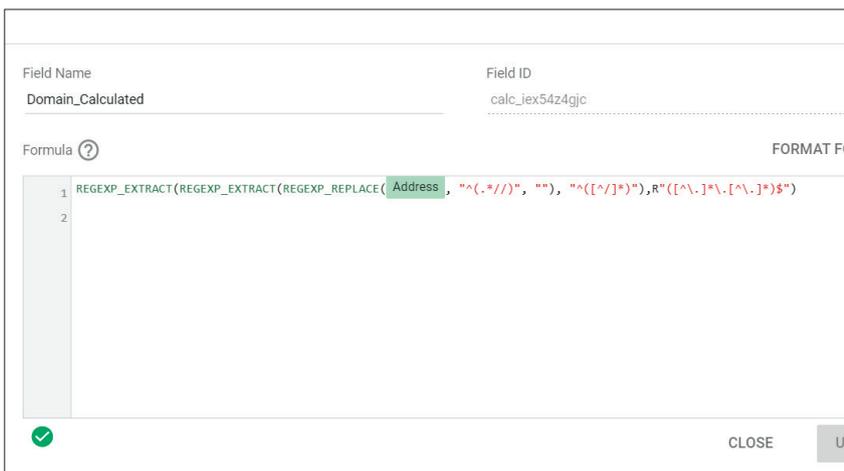


Abb. 23: Erstellung des kalkulierten Feldes

### Der Inhalt des Reports

Die erste Seite des Reports enthält fünf Seiten mit dem schnellsten TTFB-Werten. Zusätzlich ist ein Dropdown eingebaut, um jede Domain mit jeder zu vergleichen und den Vergleich grafisch darzustellen. Die zweite Seite des Reports enthält eine Tabelle mit allen Teilnehmern des Monitorings. Sie stellt TTFB von heute dar und errechnet den prozentualen Unterschied zum Vortag.

### Ein kalkuliertes Feld zur Herauslösung von Domains aus den URLs

Zusätzlich wurde in den Report ein sog. kalkuliertes Feld (calculated field) eingebaut, das aus einer URL eine Domain samt TLD-Endung herauslöst – sogenannte „naked Domains“ lassen sich gut darstellen.

Das kalkulierte Feld lässt sich auf folgende Weise erstellen: In der rechten Spalte „Available Fields“ ganz unten gibt es einen Button „ADD A FIELD“. Dieser wird angeklickt und in dem neuen Fenster wird Folgendes eingegeben. Es wird ein Field-Name vergeben. In das Feld „Formel“ wird der reguläre Ausdruck eingegeben, der aus einer URL die Domain herauslöst. Im aktuellen Fall lautet der reguläre Ausdruck wie folgt:

```
REGEXP_EXTRACT(REGEXP_EXTRACT(REGEXP_REPLACE(Address, "(.*)//", ""), "(.*)"), "(.*)")
```

Der reguläre Ausdruck bezieht sich auf die Spalte „Address“, in welcher die URLs gespeichert sind. Abbildung 22 zeigt, wie die Erstellung des kalkulierten Feldes aussieht.

Zwar lassen sich Domains aus URLs noch in Google Sheets mit einer Formel herauslösen, in GDS ist die Lösung mit dem kalkulierten Feld smarter.

Der vorkonfigurierte Report steht für alle kostenlos unter dem Link <http://einfach.st/froggds2> zu Verfügung. Fragen dazu beantwortet der Autor gerne. ¶