



Mario Fischer, Patrick Lürwer

R-LEUCHTUNGEN!

Teil 4: Daten direkt von Google absaugen

DER AUTOR



Mario Fischer ist Herausgeber und Chefredakteur der Website Boosting und seit der ersten Stunde des Webs von Optimierungsmöglichkeiten fasziniert. Er berät namhafte Unternehmen aller Größen und Branchen und lehrt im neu gegründeten Studiengang E-Commerce an der Hochschule für angewandte Wissenschaften in Würzburg.

DER AUTOR



Patrick Lürwer ist Senior Analyst bei get.traction GmbH. Dort ist er für die Datenerfassung, -aufbereitung und -analyse zuständig. Sein tägliches Handwerkzeug sind R, Python und KNIME.

In den Ausgaben bis 54 bis 58 der Website Boosting konnten Sie in der Serie „R4SEO“ von Patrick Lürwer nachvollziehen, wie man die kostenlose Software R verwendet, was sie leistet und wie man sie nutzbringend für die eigene Arbeit für SEO bzw. die Aufklärung im Online-Marketing einsetzen kann. R wurde ja ursprünglich für Statistik entwickelt. Wer das bisher als Entschuldigung verwendet hat, es deswegen links liegen zu lassen, dem sei versichert, dass er damit komplett falschliegt.

R kann für den Einsatz im Unternehmen, allen Bereichen voran „Online“, sehr viel mehr leisten, als statistische Berechnungen durchzuführen. Genau genommen ist es ein wirklich nützliches Helferlein bei allen Aufgaben im Umgang mit größeren Datenmengen, mit Daten, die man erst in eine gewisse Struktur bringen muss, und bei der automatischen oder halb automatischen Datenbeschaffung aus praktisch fast allen Quellen aus dem Web!

Für die interessierten Einsteiger, aber auch für alle, die nach der Serie von Patrick Lürwer „R-Blut“ geleckt haben, startete in der Ausgabe 62 die neue anwendungsorientierte Serie „R-Leuchtungen“. Sie werden in jeder Ausgabe erfahren, wie sie ohne Programmierkenntnisse jeweils ein definiertes und in der Online-Praxis häufiger auftretendes Problem rund um das Thema Daten und Auswertungen lösen können. Und keine Sorge, die kleinen Hilfe-Tutorials nehmen Sie Schritt für Schritt an der Hand, sodass Sie auch als Neuling von der Power von R profitieren können. Was hält Sie also ab, das einfach mal auszuprobieren? Die einzelnen Schritte müssen Sie übrigens nicht im Detail verstanden haben. Um an die wertvollen Rankingdaten von Google en bloc zu kommen, müssen Sie nur nachmachen, was Sie hier beschrieben finden.

Diesmal geht es darum, direkt relevante Daten „am Stück“ aus der Google Search Console zu ziehen und sie mit Daten aus Google Analytics zu verknüpfen. Einmal eingerichtet, können Sie dieses Skript bei Bedarf immer wieder starten, wenn Sie frische Daten benötigen. Und das Beste: Alles ist komplett kostenlos.

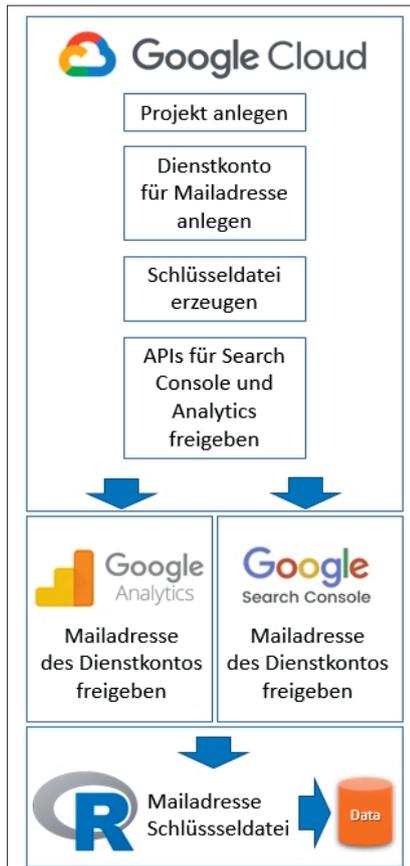


Abb. 1: Übersicht über die nötigen Schritte

Schritt 1: Bei den Google Services anmelden – das ist einmalig nötig

Es ist wie immer: Vor den Spaß hat immer irgendjemand den Schweiß gesetzt. Damit man sich über die Google APIs (Datenschnittstellen) auch wirklich Daten übertragen lassen kann, müssen zweierlei Dinge passieren. Man muss sich einen sog. (kostenlosen) Service-Account erstellen, um sich gegenüber Google digital mit dem abfragenden Tool authentifizieren zu können. Danach muss man für diesen Service-Account die betroffene eigene Domain freigeben, damit genau diese Daten auch fließen können. Wir zeigen Ihnen, wie das für die Google Search Console sowie Google Analytics genau geht. Schritt für Schritt zum Nachmachen.

Das klingt auf den ersten Blick umständlich. Wenn man mit Werkzeugen wie Screaming Frog oder den SEO-Tools für Excel bei Google abfragt, braucht man schließlich auch keinen

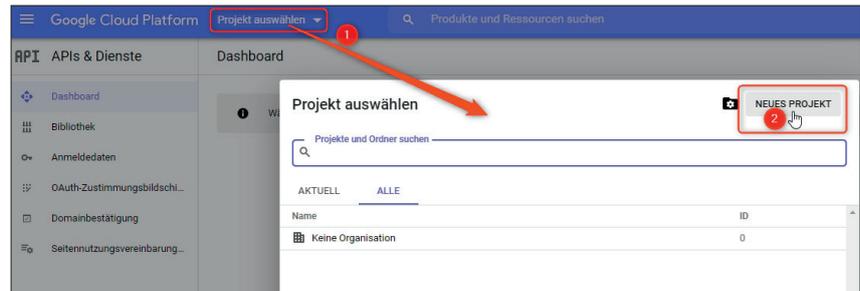


Abb. 2: Ein Projekt in der Cloud-Plattform anlegen

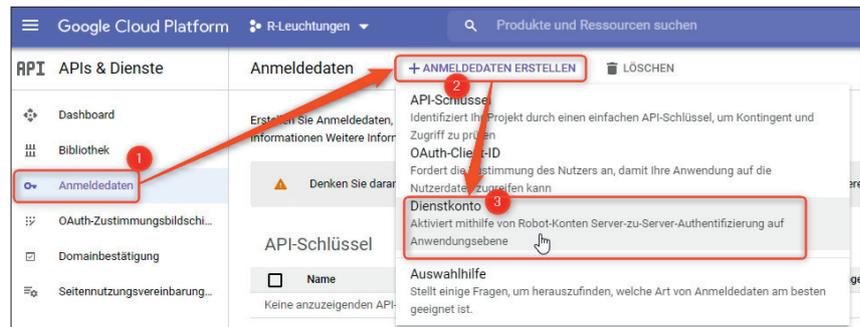


Abb. 3: Ein Dienstkonto in der Cloud-Plattform erstellen

Service-Account. Dafür poppt da jedes Mal eine Abfrage über ein Berechtigungsfenster auf. Über R macht man das eleganter und löst dies dauerhaft, indem man eine Datei mit einem digitalen Schlüssel auf dem Computer ablegt und im Skript dann einfach dieses Verzeichnis angibt. Ab dann laufen alle Abfragen ohne Nachfragen und starten sofort und bei Bedarf vollautomatisch. Als weiterer Vorteil ist zu nennen, dass später auch verschiedene Skripte einfach auf diese Datei zugreifen können. Der Aufwand für die Erstellung eines solchen Service-Accounts ist zwar einmalig etwas aufwendiger und wirkt vielleicht auch ein wenig abschreckend, aber dafür spart man hinterher deutlich Zeit und die Möglichkeiten zur Datenverarbeitung mit Google-Daten sind praktisch unbegrenzt.

Die Google-Cloud-Plattform erstmalig öffnen

Rufen Sie <https://console.cloud.google.com/apis/dashboard> und melden Sie sich mit Ihrem bzw. einem Google-Account an. Hierzu reicht es, eine Gmail-Adresse samt Kennwort zum Einloggen zu haben. Genau diese beiden Daten (Mailadresse und Kenn-

wort) können Sie hier auch verwenden. Bestätigen Sie die Datenschutzhinweise und ggf., ob Sie von Google Mails zur Cloud-Plattform haben möchten. Und schon sind Sie „drin“.

Im Dashboard (Navigation links) klicken Sie oben auf Projekt auswählen (Ziffer 1 in Abbildung 2) und anschließend in dem aufgehenden Fenster auf „NEUES PROJEKT“ rechts oben (Ziffer 2). Suchen Sie sich einen beliebigen Projektnamen aus, er dient nur zur Übersicht. Das Fenster „Speicherort“ unter dem Projektnamen können Sie getrost auf „Keine Organisation“

INFO

Wer R noch nicht installiert hat, kann das schnell und einfach bewerkstelligen. Unter cran.r-project.org/bin/ findet man die aktuellen Versionen für Windows (32/64 Bit), MacOS (X) und Linux. Dort im Unterverzeichnis „base“ liegen dann die installierbaren Dateien. Nach der Installation holt man sich am besten gleich unmittelbar die Software „R-Studio“, die sich als eine Art Hülle um den Kern von R als grafische Benutzeroberfläche legt und viele Tools und Eingabehilfen zur Verfügung stellt. Die Bedienung von R wird mit R-Studio fast zum Kinderspiel. Man findet sie unter: rstudio.com/products/rstudio/download/

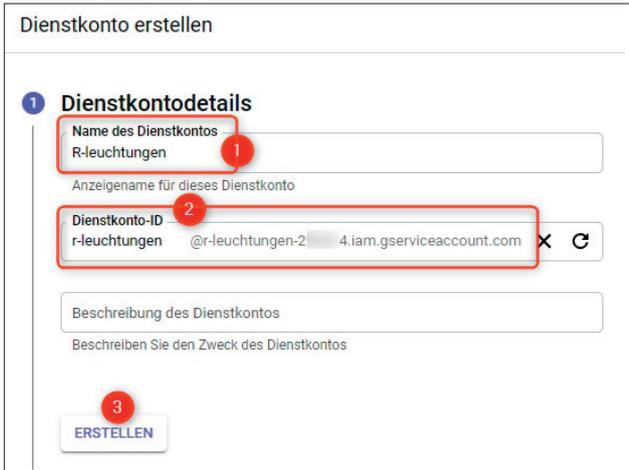


Abb. 4: Das Dienstkonto benennen

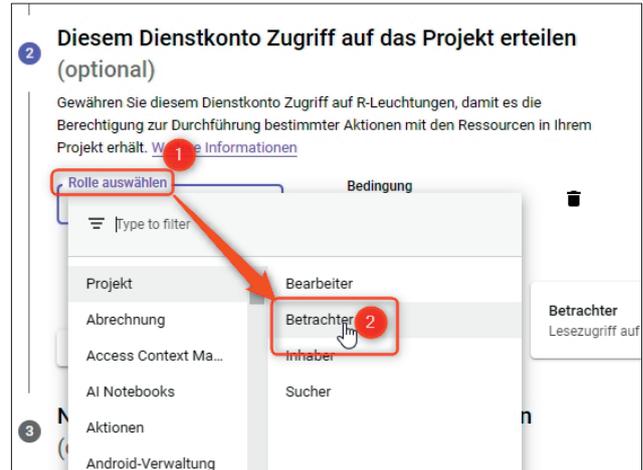


Abb. 5: Die Rechte-Rolle als „Betrachter“ genügt hier

stehen lassen. Mit dem Button „ERSTELLEN“ wird das Projekt angelegt. Das kann durchaus ein klein wenig Zeit in Anspruch nehmen, aber anschließend meldet das System sich zurück und ist automatisch im neuen (noch leeren) Projekt.

Die Schlüsseldatei erzeugen

Anschließend wechseln Sie in der linken Hauptnavigation von „Dashboard“ zu „Anmeldedaten“ (Abbildung 3, Ziffer 1). Nun klicken Sie rechts oben auf „+ANMELDEDATEN ERSTELLEN“ (Ziffer 2) und in dem jetzt aufklappenden Fenster auf „Dienstkonto“ (engl. Service Account, Ziffer 3).

Anschließend vergeben Sie wieder einen möglichst sprechenden Namen Ihrer Wahl im Feld „Name des Dienstkontos“ (Abbildung 4, Ziffer 1). Im Feld darunter (Ziffer 2) sehen Sie gleich, wie sich die Mailadresse des Dienstkontos zusammensetzt. Die braucht man später, um darauf zugreifen zu können, bzw. diese Mailadresse wird in der Search Console und Google Analytics als Nutzer angelegt. Vergeben Sie also einen Namen, mit dem Sie später auch noch etwas anfangen können. Sonst fragen Sie sich ggf. in ein paar Jahren, was das für ein komischer Maileintrag ist. Mit einem Klick auf „ERSTELLEN“ (Ziffer 3) schließen Sie auch diesen Schritt ab.

Danach geben Sie bei „Rolle auswählen“ (Abbildung 5) noch „Betrachter“ an, da die Daten ja nur abgeholt

werden, aber nichts zurückgeschrieben wird. Anschließend klicken Sie den „FERTIG“-Button an.

Sie sehen nun wieder die Ansicht „Anmeldedaten“. Allerdings finden Sie jetzt in der dritten Datenreihe unten unter „Dienstkonten“ die eben angelegten Anmeldedaten bzw. die aus Ihrem gewählten Namen erzeugte E-Mail-Adresse.

Sie sieht vom Muster in der Regel etwa so aus: *ihrgewahltername@ihrgewahltername.iam.gserviceaccount.com*. Hinweis: Bitte kopieren bzw. speichern Sie sich diese Mailadresse an einen Ort, wo Sie sie in ein paar Minuten nachher wiederfinden bzw. abholen. Klicken Sie auf diesen Eintrag und Sie gelangen in die Dienstkontodetails. Dort finden Sie unter „Schlüssel“ (Abbildung 6) die Funktion „Schlüssel

hinzufügen“ (Ziffer 2). Im aufklappenden Auswahlmenü wählen sie „Neuen Schlüssel erstellen“ (Ziffer 3). Es öffnet sich ein weiteres Fenster, in dem man den Schlüsseltyp „JSON“ auswählt (Abbildung 7).

Anschließend geht der Datei-speichern-unter Dialog Ihres Computers auf. Speichern Sie den eben erzeugten Schlüssel am besten in einem neuen, dafür angelegten Verzeichnis. Geht der Schlüssel verloren, kann er nicht mehr reproduziert werden und Sie müssen einen neuen generieren! Das ist kein Beinbruch, aber vermeidbarer Aufwand. Merken Sie sich also bitte, wohin Sie den Schlüssel speichern. Wenn Sie R schon installiert haben, legen Sie ihn der Einfachheit halber einfach im R-Verzeichnis ab, dann muss kein gesonderter Verzeichnis angegeben werden. Ach



Abb. 6: Einen digitalen Schlüssel erstellen

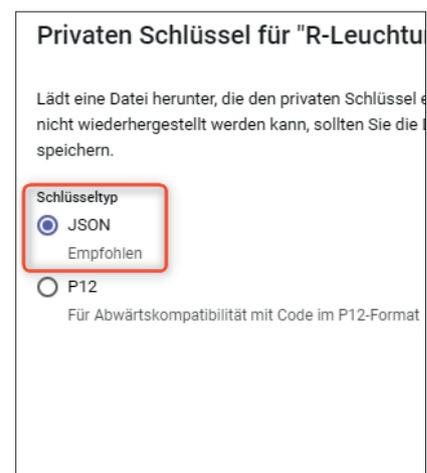


Abb. 7: Schlüsselstyp ist „JSON“ -- Achtung, bewahren Sie den Schlüssel gut auf!

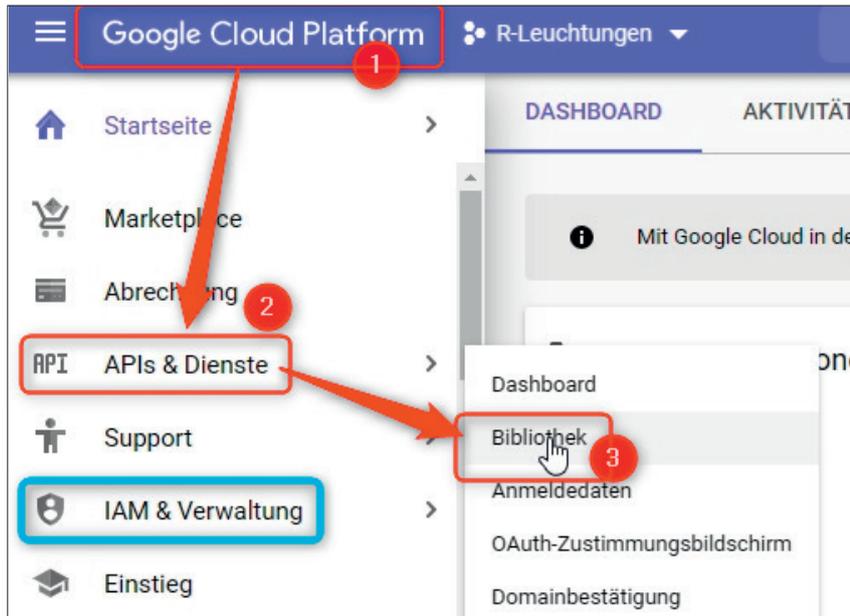


Abb. 8: Zurück ins API-Menü

ja: Geben Sie den Schlüssel nicht aus der Hand, weil man damit theoretisch Ihre eben angelegten Cloud-Ressourcen anzapfen könnte.

Wenn alles geklappt hat, sehen Sie in der Übersicht unter „Dienstkonten“ in Ihrem Projekt einen angelegten Schlüssel mit dem Status „Aktiv“. Achtung, jetzt sind Sie im Menüpunkt „IAM & Verwaltung“.

Jetzt noch die beiden Datenschnittstellen (APIs) aktivieren

Über einen Klick oben auf den Schriftzug „Google Cloud Platform“ können Sie die Hauptnavigation aufrufen und dort in den Menüpunkt „APIs & Dienste“ zurückkehren, von dem aus wir zu Anfang gestartet sind. Entweder direkt im Ausklappenmenü wie in Abbildung 8, Ziffer 2 und 3, gezeigt oder dann im Untermenü in APIs & Dienste wählen Sie als nächstes „Bibliothek“ aus.

Jetzt wird es einfach. Wie in Abbildung 9 gezeigt, tippen Sie im Suchfeld „Google Search Console“ (Ziffer 1) und anschließend erscheinen die Ergebnisse. Wählen Sie in den Ergebnissen den Eintrag „Google Search Console API“ (Ziffer 2) aus (genau diesen!) und klicken dann auf „AKTIVIEREN“ (Ziffer 3).

Genau nach dem gleichen Prinzip wird auch die API von Google Analytics aktiviert. Klicken Sie erneut oben auf „Google Cloud Platform“, dann „APIs & Dienste“ und schließlich wieder auf „Bibliothek“. Dort geben Sie im Suchfeld nun „Google Analytics API“ ein und klicken auf das passende Suchergebnis. Achtung: Verwechseln Sie im Suchergebnis nicht die auch auftauchende „Google Analytics Reporting API“. Ein Mausklick auf „AKTIVIEREN“ schließt auch diesen Schritt ab.

Das war alles, was Sie wie erwähnt nur einmal initial in der Cloud-Plattform einrichten müssen. Wenn Sie alle Schritte befolgt haben, liegen Ihnen nun eine Service-Account-E-Mail-Adresse und eine Schlüsseldatei im JSON-Format vor.

Schritt 2: Google Analytics und die Search Console für die API-Abfragen freischalten

Nachdem Sie gewissermaßen die „Datensender“, also die API-Schnittstellen der Tools, eingerichtet und aktiviert haben, müssen Sie jetzt den Anwendungen Google Analytics und der Search Console mitteilen, dass und an wen sie ihre Datenflut über diese APIs freigeben dürfen.

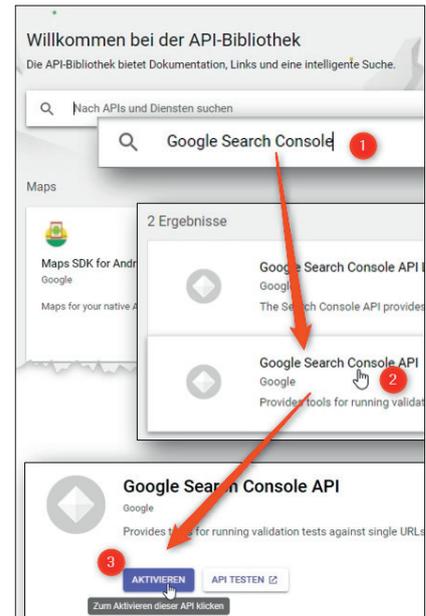


Abb. 9: Die API-Schnittstellen müssen noch aktiviert werden

Die Daten der Search Console freischalten

Rufen Sie die Google Search Console (search.google.com/search-console) auf (die sie natürlich für Ihre Domain(s) bereits eingerichtet haben müssen) und wählen Sie dort die betroffene Property (in der Regel die Domainadresse) links oben in der Auswahlbox auf (Abbildung 10, Ziffer 1). Anschließend rufen Sie unten im linken Menü die „Einstellungen“ auf (Ziffer 2). Nun legen Sie unter „Nutzer und Berechtigungen“ über den blauen Button rechts oben „NUTZER HINZUFÜGEN“ (Ziffer 3) den neuen Nutzer an. Dieser besteht aus der vorhin generierten E-Mail-Adresse, die Sie sich für diesen Zweck ablegen sollten (in unserem Beispiel oben war das `ihrgewahltername@ihrgewahltername.iam.gserviceaccount.com`). Als Berechtigung können Sie die Voreinstellung „Uneingeschränkt“ stehen lassen, da es ja Ihr Account ist. Nach dem Bestätigen durch „HINZUFÜGEN“ (Ziffer 5) erscheint die Mail in der Liste der Nutzer (wo sie später bei Bedarf auch wieder gelöscht werden könnte, wenn man sie nicht mehr benötigt oder haben möchte).

Die Daten von Google Analytics freischalten

Die Freischaltung unseres neuen Nutzers in Google Analytics (*analytics.google.com/analytics/web*) funktioniert nach genau demselben Prinzip. Sie rufen Google Analytics auf und öffnen die Datenansicht, von der Sie später die Daten abfragen möchten. Links unten klicken Sie auf „Verwaltung“ und dort in der mittleren Spalte auf „Property – Nutzerverwaltung“. Es erscheint eine Liste mit den bisherigen Zugriffsberechtigten. Ein Klick rechts oben auf den blauen Kreis mit dem Pluszeichen fügt einen neuen Nutzer hinzu. Analog dem Vorgehen in der Search Console tragen Sie hier die gleiche generierte E-Mail-Adresse wie eben ein und vergeben als Berechtigung „Lesen und analysieren“ durch Anklicken des Kästchens. Das genügt für den Datenabruf. Nach Abschluss erscheint auch hier der neue Nutzer bzw. die eben eingetragene Mailadresse als ab jetzt berechtigter Nutzer.

Das war alles, was Sie für die Vorbereitung – wie gesagt: einmalig – tun mussten. Jetzt geht es direkt in R und an das Datenzapfen.

Schritt 3 – endlich: Die Daten über R holen und aufbereiten

Wenn Sie einmal an Google Analytics und die Search Console „angebunden“ sind, müssen Sie sich ab dann um diesen Punkt nicht mehr kümmern. Ab jetzt können Sie mit beliebigen Anwendungen/Skripten in R die nötigen Daten von Google abholen, aufbereiten, mischen, abgleichen, auf Anomalien prüfen und vieles mehr. Im Lauf der Zeit werden Sie hier im Heft immer wieder nützliche Skripte finden, die Sie einfach nur kopieren müssen und in R einfügen, ohne dass Sie selbst programmieren oder im Detail verstehen müssen, was genau im Hintergrund abläuft.

Da wir schon im vierten Teil dieser

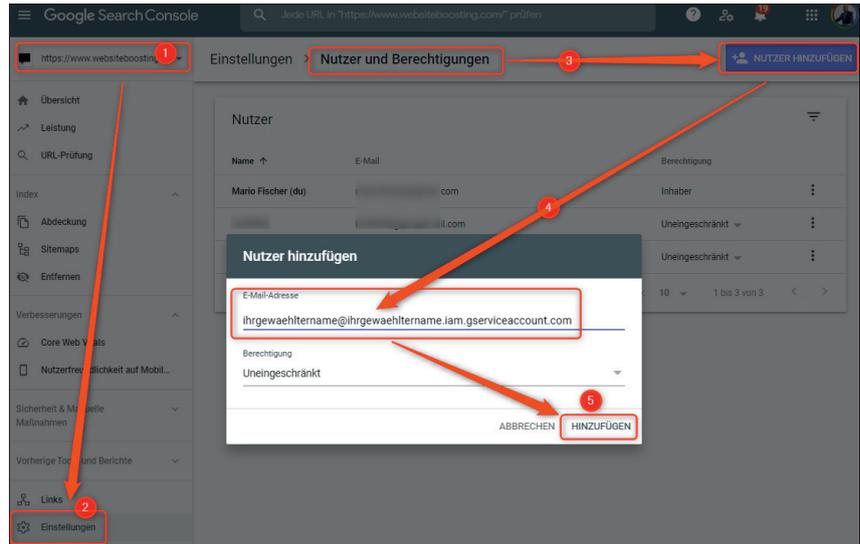


Abb. 10: Das Eintragen des neuen „Benutzers“ in der Search Console ist einfach zu bewerkstelligen

Serie sind und auch in den Heften vorher (viele Artikel sind mittlerweile kostenlos online auf *www.websitboosting.com* verfügbar) schon einige Beiträge zu R hatten, gehen wir hier davon aus, dass R bzw. eben auch R-Studio als vernünftige Bedienungsfläche für R installiert ist.

Öffnen Sie also R-Studio und legen Sie zunächst ein neues Projekt (File/New Project) an. Bitte beachten Sie: Im Gegensatz zu den vorherigen Skripten lässt sich das Folgende nicht in einem Stück ausführen. Denn sie müssen unter anderem die abzufragende Property und den Datumszeitraum definieren. Kopieren sie daher den Code in eine eigene Skript-Datei (File/New File/R Script) und führen Sie die einzelnen Code-Blöcke nach und nach aus. Dazu markieren Sie den jeweils auszuführenden Code-Block mit der Maus und drücken STRG + ENTER.

Diesmal benötigen wir drei neue Bibliotheken zusätzlich zu „tidyverse“, die wir die letzten Male ja schon geladen haben. Die Namen sind eigentlich selbstsprechend. Sie dienen zur Authentifizierung bei Google, für die Search Console und für Google Analytics:

- » `library(tidyverse)`
- » `library(googleAuthR)`
- » `library(searchConsoleR)`
- » `library(googleAnalyticsR)`

Die zweite Bibliothek übernimmt den gesamten Prozess der Authentifizierung und wir müssen nur die Rechte angeben, die wir R geben. In unserem Fall reichen Leserechte (readonly) völlig aus. Wer mehr über die zu vergebenden Rechte nachlesen möchte, kann das hier <http://einfach.st/rgscapi> bei Google tun.

```
options(googleAuthR.scopes.selected = c(„https://www.googleapis.com/auth/webmasters.readonly“, „https://www.googleapis.com/auth/analytics.readonly“))
```

Jetzt müssen noch Name und Speicherort der Schlüsseldatei angegeben werden. Wenn sie ins Arbeitsverzeichnis von R gespeichert wurde, muss natürlich kein Pfad angegeben werden. Die Befehlszeile lautet dann z. B.:

```
gar_auth_service(„r-leuchtung-a7a46a613ad4.json“)
```

Natürlich müssen Sie den Dateinamen zwischen den Anführungszeichen durch Ihren eigenen Dateinamen ersetzen.

Rechts unten in R-Studio sehen Sie, sofern Sie nichts verändert haben, das aktuelle Arbeitsverzeichnis (Abbildung 11). Hier sollte die Schlüsseldatei mit dem oben beschriebenen Namensmuster liegen. Sie enthält Ihren sog.

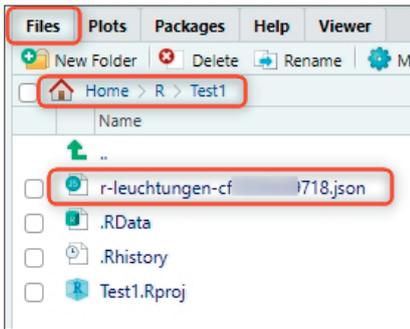


Abb. 11: Unter „Files“ lässt sich kontrollieren, ob die Schlüsseldatei da ist

„Private Key“, mit dem Sie sich Google gegenüber als rechtmäßig für den Datenabzug ausweisen. Sollten Sie die Datei in einem anderen Verzeichnis abgelegt haben, können Sie das natürlich tun – wenn Sie mit Windows arbeiten, müssen Sie aber dann diesen Pfad (mit einem „Forwardslash“, also „/“ statt „\“) angeben. Das sieht dann z. B. so aus:

```
gar_auth_service(„C:/Verzeichnis1/SammlungX/r-leuchtung-a7a46a613ad4.json“)
```

Beim erstmaligen Ausführen des Skriptes erscheint eventuell auf der Console die Frage, wo das „Auth-Object“ gespeichert werden soll. Akzeptieren Sie dann am besten der Einfachheit halber die Vorauswahl bzw. den Vorschlag von R.

Hat alles geklappt bisher? Das können Sie ganz einfach prüfen, wenn Sie

```
list_websites() %>%  
View()
```

eingeben. Links oben erscheint eine kleine Tabelle mit den Einträgen der URL bzw. der Domain und dem Zugrifflevel, wo „siteFullUser“ eingetragen ist (Abbildung 12).

Jetzt legen Sie die sog. Property in eine Variable, was den späteren Umgang erleichtert:

```
GSC_PROP <- „https://www.ihre-domain.de/“
```

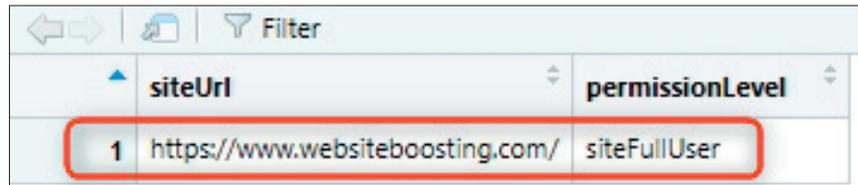


Abb. 12: Der Zugriff auf die Search Console ist da!

Achten Sie darauf, dass Sie den Trailingslash der Property nicht vergessen!

Daten aus der Search Console holen

Natürlich müssen wir der Search Console sagen, aus welchem Zeitraum wir Daten haben möchten. Das geht mit den folgenden zwei Zeilen Code, wobei Sie natürlich das Datum nach Ihren Wünschen anpassen sollten. Bitte denken Sie immer daran, dass wir bei der Search Console eine Verzögerung von drei Tagen haben. Das bedeutet, bei „END_DATE“ können Sie zwar das aktuelle Datum angeben und das akzeptiert die API. Allerdings werden Sie für die letzten drei Tage dann natürlich keine Daten bekommen.

```
START_DATE <- „2020-09-01“  
END_DATE <- „2020-09-30“
```

Und jetzt können wir Daten holen! Dazu dient der nachfolgende Code, der für den definierten Zeitraum die Kombination aus Seite und Keyword sowie dazu die Klicks, Impressions, CTR und die Rankingposition holt. Da man per Standard nur 1.000 Daten zurückbekommt, setzen wir „rowLimit“ vorsichtshalber auf 100.000. Wie in Abbildung 13 zu sehen ist, kommen auf den ersten Schwung (Ziffer 1) 25.000 Datenreihen und es werden automa-

tisch mehrere sog. „Calls“ hintereinander gemacht. Der zweite Schwung hat weitere 6.392 Datenreihen bekommen, und da jetzt nichts mehr da ist bzw. alle hier im Beispiel vorhandenen 31.392 (25.000 + 6.392) Datensätze übermittelt wurden, bricht die Übertragung mit einer ignorierbaren Warnmeldung (No data found) ab.

```
gsc_data <- search_analytics(  
  siteURL = GSC_PROP,  
  startDate = START_DATE,  
  endDate = END_DATE,  
  dimensions = c(„page“,  
  „query“),  
  searchType = „web“,  
  rowLimit = 100000)
```

Wenn Sie jetzt die im Speicher erzeugte Datentabelle ansehen möchten, führen Sie einfach

```
gsc_data %>%  
View()
```

aus. Und schon erscheint rechts oben ein Ausschnitt mit den Werten. Der Befehl

```
write_csv(gsc_data, „gsc_data.csv“)
```

schreibt ein CSV-File mit dem Namen gsc_data ins Arbeitsverzeichnis. Und dort zeigt sich dann, dass in

```
> gsc_data <- search_analytics(siteURL = GSC_PROP,  
+                             startDate = START_DATE,  
+                             endDate = END_DATE,  
+                             dimensions = c(„page“, „query“),  
+                             searchtype = „web“,  
+                             rowlimit = 100000)  
Fetching search analytics for url: https://www.websiteboosting.com/ dates: 2020-09-01 2020-09-30 dimensions: page query dimensionFilterExp: searchtype: web aggregationType: auto  
Batching data via method: bybatch  
with rowLimit set to 1e+05 will need up to [5] API calls  
Page [1] of max [5] API calls  
Downloaded 25000 rows  
Page [2] of max [5] API calls  
Downloaded 6392 rows  
Page [3] of max [5] API calls  
Warnmeldung:  
No data found for supplied dates - returning NA  
> |
```

Abb. 13: Sieht dramatisch aus, aber alles ist in bester Ordnung ...

diesem Beispiel tatsächlich alle 31.392 Suchanfragen komplett und ohne Einschränkung mit allen zugehörigen Metriken übertragen wurden, wie Abbildung 14 zeigt.

query	click	impression	ctr	position
...tracking-beacon	2	8	0,25	1,25
...gemahnt-ueber-google-und-a	2	27	0,07	1,26
...wirkungen-der-amazon-verka	4	27	0,15	1,30
...test-1.html	0	106	0,00	1,30
...amazon-alexa-marketing	0	3	0,00	1,33
...suchsysteme	6	42	0,14	1,40
...youtube-statistiken-ansehen	14	110	0,13	1,45
...google-shopping-traffic	0	11	0,00	1,45
...pagerank	1	12	0,08	1,10

Abb. 14: Von wegen „not provided“ – so bekommen Sie ALLE Suchbegriffe Ihrer Domain

Daten aus Google Analytics holen

In Google Analytics gibt es ja bekanntlich verschiedene Datenansichten, die man einzeln auswählen bzw. ansprechen kann. Am einfachsten ist es, sich gleich in R über die Codezeile

```
ga_account_list() %>%
  View()
```

die wählbaren Datenansichten bzw. deren Nummern ausgeben zu lassen. Links oben in R-Studio erscheinen nach der Eingabe alle Daten zum Analytics-Account. Im Beispiel (Abbildung 15) ist nur eine Datenansicht vorhanden, also gibt es hier auch nur eine ID zur Auswahl.

Diese bzw. die ID, die Sie haben möchten, schreiben Sie mit folgendem Befehl der Variable GA_VIEW_ID zu. Natürlich setzen Sie Ihre Zahl ein und nicht die hier abgedruckte.

```
GA_VIEW_ID <- „123456789“
```

Sie können diese ID aber ebenso vorab in Ihrem Google-Analytics-Account einsehen (Abbildung 16). Am besten kopieren Sie sich dort diese und ggf. weitere IDs für weitere Datenansichten heraus, damit Sie diese später flexibel im Skript einsetzen können!

Start- und Enddatum wurde weiter oben ja schon definiert, das muss also nicht erneut gemacht werden. Jetzt sind nur noch ein paar Filtersetzungen nötig, damit Google Analytics weiß, was wir genau geliefert haben wollen. In diesem Fall wird auf organische Besuche gefiltert, d. h., es wird nur Traffic ausgewiesen, der über den unbezahlten Kanal von Suchmaschinen gekommen ist.

```
channel_filter <- dim_filter(dimension = „channelGrouping“,
  operator = „EXACT“,
  expressions = „Organic Search“)
filter_clause <- filter_clause_
ga4(list(channel_filter))
```

Anschließend erfolgt die eigentliche Datenabfrage:

```
ga_data <- google_analytics(viewId = GA_VIEW_ID,
  date_range = c(START_DATE, END_DATE),
  metrics = c(„sessions“, „bounceRate“),
  dimensions = „landingPagePath“,
  dim_filters = filter_clause,
  anti_sample = TRUE)
```

Eine kurze Kontrolle ermöglicht wieder der View-Befehl, diesmal für den Datensatz „ga_data“:

```
ga_data %>%
  View()
```

Links oben erscheinen drei Spalten mit der URL („landingPagePath“), der Anzahl Sessions für den Zeitraum und der Bouncerate.

Leider hat Analytics die unangenehme Eigenschaft, die Domainbezeichnung wegzulassen und nur die Pfade auszuweisen, also z. B.: /magazin/33/youtube-analytics.html statt https://www.websiteboosting.com/magazin/33/youtube-analytics.html

Wenn man die Daten von Analytics mit denen der Search Console matchen

webPropertyName	viewId	viewName
WebsiteBoosting.com	135134783	Alle Websitedaten

Abb. 15: Die ID der Datenansicht(en)

möchte, muss das noch ergänzt werden, damit die Einträge bei den URL wirklich identisch sind. Das geht wie folgt:

```
DOMAIN <- „https://www.ihr-domain-name.de“
ga_data <- ga_data %>%
  mutate(landingPagePath =
    pasteO(DOMAIN, landingPagePath))
```

Achtung: Schreiben Sie keinen Schrägstrich hinter den Domainnamen, also nicht https://www.ihr-domainname.de/, sonst stehen da beim Matching zwei // fehlerhaft in der URL.

Eine CSV Datei names „ga_data“ erzeugen wir bei Bedarf wieder mit

```
write_csv(ga_data, „ga_data.csv“)
```

Die Datenhochzeit der Search Console und Google Analytics

Jetzt fehlt nur ein Befehl, der beide Datensets miteinander vereinigt:

```
gsc_ga_data <- gsc_data %>%
  full_join(ga_data, by = c(„page“ = „landingPagePath“))
```

Für jede URL, die in beiden Datensets vorhanden ist (das muss ja nicht immer der Fall sein), werden die Daten ergänzt, sodass Sie jetzt eine komplette Tabelle zur Verfügung haben, die folgende Spalten mit Daten enthält:



Abb. 16: Hier finden Sie die IDs Ihrer Datenansichten in Google Analytics

url	query	clicks	impressions	ctr	position	sessions	bounceRate
https://www.websiteboosting.com/	website boosting	12.388	21.964	56%	1,15	21.080	14,8
https://www.websiteboosting.com/	websiteboosting	3.572	7.410	48%	1,00	21.080	14,8
https://www.websiteboosting.com/abonnemL.html	website boosting abo	1.026	1.406	73%	1,00	1.666	20,4
https://www.websiteboosting.com/magazin/40/aktives-bewertungsmanagement	bewertungsmanagement	988	6.042	16%	1,03	1.224	22,2
https://www.websiteboosting.com/magazin/33/youtube-analytics.html	youtube kanal statistiken	950	5.928	16%	3,57	5.202	65,4
https://www.websiteboosting.com/magazin/33/youtube-analytics.html	youtube analytics	912	148.048	1%	7,03	5.202	65,4
https://www.websiteboosting.com/magazin/37/personal-branding-so-bauen-s	personal branding	874	81.624	1%	10,36	1.972	15,5
https://www.websiteboosting.com/magazin/37/youtube-analytics.html	youtube analytics von anderen	760	2.280	33%	1,82	5.202	65,4
https://www.websiteboosting.com/	website boosting magazin	722	2.660	27%	1,01	21.080	14,8
https://www.websiteboosting.com/magazin/45/10-tipps-fuer-mehr-traffic-dar	mehr traffic auf website	722	2.774	26%	1,62	1.564	8,7
https://www.websiteboosting.com/magazin/52/seo-tools-teil-9-screaming-fro	screaming frog	722	101.422	1%	13,49	1.564	6,5
https://www.websiteboosting.com/magazin.html	website boosting	684	21.964	3%	1,15	1.054	16,1
https://www.websiteboosting.com/magazin/35/google-analytics-fuer-einsteig	google analytics für anfang	646	3.344	19%	1,61	1.598	19,1
https://www.websiteboosting.com/magazin/44/berechnen-sie-ihren-pagerank	berechnen	570	2.660	21%	3,00	2.482	21,9
https://www.websiteboosting.com/magazin/33/youtube-analytics.html	youtube statistiken ansehen	532	4.180	13%	1,45	5.202	65,4
https://www.websiteboosting.com/magazin/39/die-produktsuche-auf-mobilg	amazon suchfeld zu klein	532	6.346	8%	5,10	1.870	45,5
https://www.websiteboosting.com/magazin/39/neu-die-google-analytics-360-	google analytics 360 kosten	494	1.862	27%	1,94	1.836	14,8
https://www.websiteboosting.com/magazin/42/neuronale-netze-verstehen-w	neuronales netz beispiel	494	5.282	9%	4,73	2.856	36,9
https://www.websiteboosting.com/magazin/49/seo-tools-unter-der-lupe-teil-	semrush	494	192.014	0%	6,06	510	6,7
https://www.websiteboosting.com/magazin/ausgabe-48-inhalt/urls-effizient-v	url weiterleitung	494	5.092	10%	5,57	3.774	40,5

Abb. 17: Fertig: kombinierte Daten via Excel aufgehübscht

URL, Keyword, Klicks, Impressions, CTR, Position, Sessionanzahl und Bouncerate

Die Anzeige der Tabelle direkt in R erzeugt man, hier nach Page und dann absteigend nach Klicks sortiert, mit:

```
gsc_ga_data %>%
  arrange(page, desc(clicks)) %>%
  View()
```

Und auch hier gelingt der Export aller Daten mit einer einzigen Zeile:

```
write_csv(gsc_ga_data, „gsc_ga_data.csv“)
```

Im Arbeitsverzeichnis liegt dann die CSV-Datei gsc_ga_data.csv für eine weitere Bearbeitung (Abbildung 17) bereit.

Fazit

Auf den ersten Blick sieht alles auf dem Papier vielleicht abschreckend kompliziert aus. Ist es aber gar nicht, wenn man es Schritt für Schritt nachvollzieht. Aber das müssen Sie gar nicht. Alle Beschreibungen hier dienen dazu, Interessierten zu erklären, was in

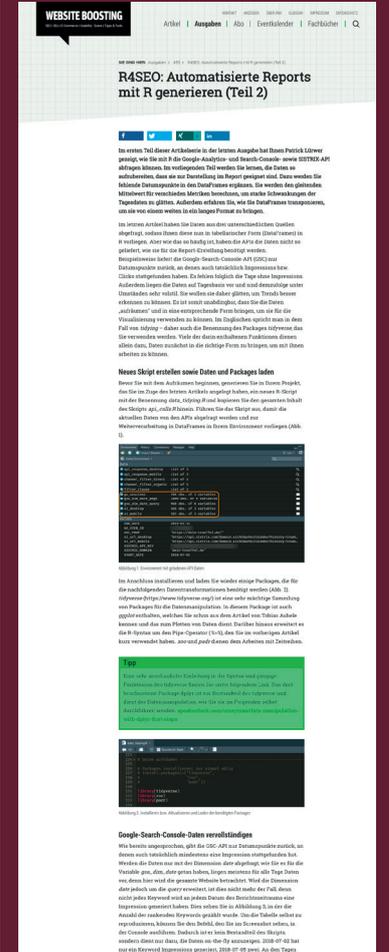
welchem Schritt eigentlich passiert.

Sie können die Power des gesamten Skriptes auch auf einmal nutzen, indem Sie es einfach in R einkopieren und die nötigen Einträge unter den mit „#### TOEDIT“ gekennzeichneten Sektionen machen. Dann lassen Sie alles auf einmal laufen und holen sich am Ende einfach nur die fertige Datei aus dem Arbeitsverzeichnis. So können sie ganz einfach – nachdem sie die abzufragende Property resp. Datenansicht definiert haben – ad hoc die Datumsvariablen für den Berichtszeitraum anpassen, um immer die neusten oder historische Daten abzufragen.

Was halten Sie da eigentlich in den Händen? Eine komplette Liste aller im definierten Zeitraum bei Google eingegeben Keywords, sofern Ihre Domain als potenzielles Ergebnis ausgewiesen wurde (Impressions), mit allen Metriken der Search Console und wichtigen Besuchsmetriken aus Google Analytics. „Not provided“ bzw. unterdrückte Keywords gehören fortan bei Ihnen zum alten Eisen. Jetzt haben Sie die volle Transparenz! ¶

TIPP

Wer Lust bekommen hat, tiefer in R einzusteigen, dem sei die fünfteilige Serie von Patrick Lürwer in der Website Boosting, Ausgaben 54 bis 58, empfohlen. Diese sind bereits frei online und als PDF-Version verfügbar.



Zur Auswahl für die kostenlosen Ausgaben (derzeit Ausgabe 1–60) kommen Sie direkt über www.websiteboosting.com/magazin.html.