Kanton Zürich
Staatskanzlei

Status Codes und Zielseiten von Subdomains

Michael Schaffner, Web-Analyst ZHweb

Ausgangslage

- Gegeben: Liste mit 3’000 Subdomains in der Form «bsp.zh.ch»

- Gesucht: Statuscode und Ziel

Lösung: 3 Funktionen

- Zieldomain abrufen: http_domains()
- Status-Code abrufen: check_url_response()
- Ziel-URL abrufen: catch_ziel_url()

Status-Code, Ziel-Domain und Ziel-Url ergeben zusammen ein Bild.

Der Umweg

- Zuerst war die Frage, ob die Subdomain weitergeleitet wird.

Weiterleitung feststellen

- http_domains_changed()

- http_domains_changed() stammt von https://www.r-bloggers.com/2018/11/using-httr-to-detect-https-redirects/

http_domains_changed()

- Nutzt GET() {httr}
- Beim httr Paket werden redirects automatisch verarbeitet.
- Man kann den Prozess aber rauslesen.

http_domains_changed()

- Im response Objekt von GET() gibt es zwei items, «headers» und «all_headers».

- «headers» enthält infos zur letzten response
- «all_headers» enthält infos zu allen responses

http_domains_changed()

- Die Aufgerufenen URLs stehen als «location» in «headers» und «all_headers» im response objekt von GET() {httr}

- Die Domain einer URL erhalten wir mit domain() {urltools}

http_domains_changed() Part 1

- httr::GET() gibt response die als Argument dient

```
http_domain_changed <-
  function(response){
    # get domain of origignal HTTP request
    orig_domain <- urltools::domain(response$request$url)
```

http_domains_chagned() Part 2

```
# extract location headers
location <-
  unlist(
    lapply(
      X   = response$all_headers,
      FUN =
        function(x){
          x$headers$location
        }
    )
  )

# new domains
new_domains <<- urltools::domain(location)
```

http_domains_chagned() Part 3

```
    # check domains in location against original domain
    any( !is.na(new_domains) & new_domains != orig_domain )
}
```

Ausführung

```
> http_domain_changed(httr::GET("ajb.zh.ch"))
[1] TRUE
```

Abänderung

- Aus http_domain_changed() wurde http_domains()

Was bei http_domains() fehlt

```r
# get domain of origignal HTTP request
orig_domain <- urltools::domain(response$request$url)
```

```r
# check domains in location against original domain
any( !is.na(new_domains) & new_domains != orig_domain )
```

http_domains()

```
http_domains <-
  function(response){
    tryCatch({ #damit kein unterbruch wenn error
    # extract location headers
    location <-
      unlist(
        lapply(
          X   = response$all_headers,
          FUN =
            function(x){
              x$headers$location
            }
        )
      )

    # new domain
    new_domain <<- urltools::domain(location)
    return(new_domain)
    }, error = function(e) {
      return(0)
    })
  }
```

http_domains() Ausführung

```
> http_domains(httr::GET("ajb.zh.ch"))
[1] "ajb.zh.ch" "www.zh.ch"
```

Status Codes abrufen

- Funktion «check_url_response» konnte 1:1 übernommen werden von https://sherif.io/2016/06/30/checking-links-responses-httr-r.html

Check_url_response benutzt HEAD() {httr}

```
> HEAD("zh.ch")
Response [https://www.zh.ch/de.html]
  Date: 2021-08-20 14:56
  Status: 200
  Content-Type: text/html;charset=utf-8
<EMPTY BODY>
```

Check_url_response

```r
check_url_response <- function(href) {
  cat('Checking', href, '...\n')
  tryCatch({
    check_head <- HEAD(href)
    return(check_head$status_code %>% as.integer())
  }, error = function(e) {
    return(0)
  })
}
```

Check_url_response() Ausführung

```
> check_url_response("zh.ch")
Checking zh.ch ...
 [1] 200
```

Ziel-URL abrufen

- Selbst gebaute Funktion catch_ziel_url()
- Ist aus check_url_response() entstanden.

Catch_ziel_url() nutzt GET() {httr}

```
> httr::GET("ajb.zh.ch")
Response [https://www.zh.ch/de/bildungsdirektion/amt-fuer-jugend-und-berufsberatung.html]
  Date: 2021-08-27 12:47
  Status: 200
  Content-Type: text/html;charset=utf-8
  Size: 61.6 kB
```

catch_ziel_url()

```r
catch_ziel_url <- function(subdomain) {
  cat('Checking', subdomain, '...\n')

  time_limit <- 10 #anzahl sekunden bis zeile auslassen

  setTimeLimit(cpu = time_limit, elapsed = time_limit, transient = TRUE)
  on.exit({
    setTimeLimit(cpu = Inf, elapsed = Inf, transient = FALSE)
  })
  tryCatch({
    check_url <- httr::GET(as.character(subdomain))
    return(check_url$url %>% as.character())
  }, error = function(e) {
    return(0)
  })
}
```

catch_ziel_url() Ausführung

```
> catch_ziel_url("ajb.zh.ch")
Checking ajb.zh.ch ...
[1] "https://www.zh.ch/de/bildungsdirektio
```

Staatskanzlei
24

Alternative um Zieldomain abzurufen

- Entweder feststellen ob es eine Weiterleitung gibt, das ist schwieriger.

- Oder nur Ziel-Domain abrufen (geht gleich wie URL abrufen)

Einfach Zieldomain abrufen

– domain() {urltools}

```
> domain(GET("ajb.zh.ch")$url)
[1] "www.zh.ch"
```

```
> domain(HEAD("ajb.zh.ch")$url)
[1] "www.zh.ch"
```

Ressourcen

- https://github.com/r-lib/httr
- https://sherif.io/2016/06/30/checking-links-responses-httr-r.html
- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- https://www.r-bloggers.com/2018/11/using-httr-to-detect-https-redirects/

Kontakt

– Michael Schaffner, Web Analyst ZHweb
– michael.schaffner@sk.zh.ch