

# HTTREX

Help make language-independent “reprex”-like documents while using web APIs from R.



useR! 2021

**Greg Freedman Ellis**  
**Twitter:** @GregFreedman  
**GitHub :** @gergness  
<https://github.com/gergness/httrex>

# WEB APIs, BUGS & REPREXES

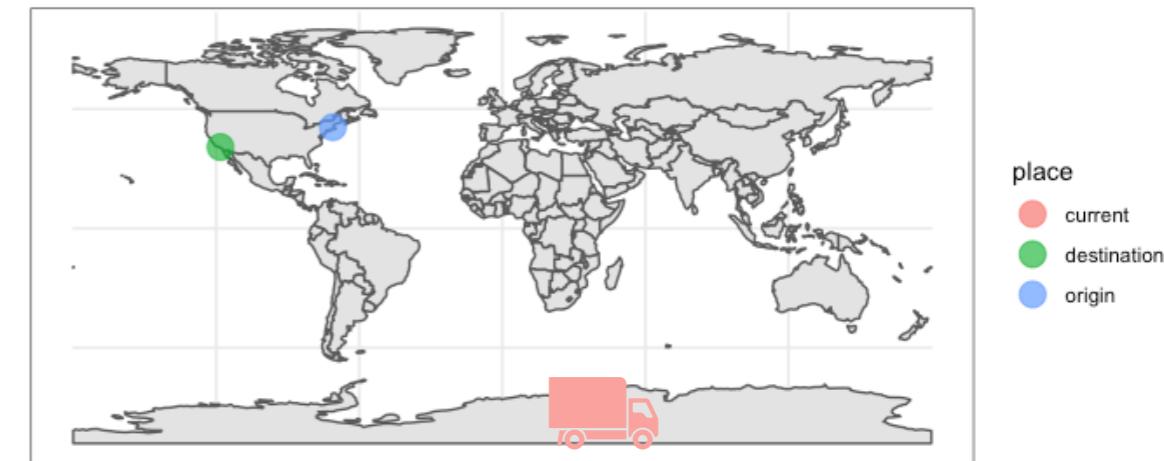
- ▶ Web APIs are a great way to collaborate with team members
- ▶ More collaboration means more places for bugs
- ▶ reprexes are a great tool for finding bugs
  - But is it really minimal if it requires someone to set up R from scratch?



# WHY IS THE TRUCK IN ANTARCTICA?

The motivation for `{httrex}`

- ▶ I was building a shiny app that showed customers where their truck currently was
- ▶ I got location data from an API built by a teammate in python
- ▶ Here's what happened:  
~~(Some details were changed to protect the innocent make a minimal example)~~



# Act 1

BOSS

<VIP> was showing your shiny app to a customer, and a few of the trucks were in Antarctica.

ME

Oh no! I can't ask <VIP> for a reprex, so this is gonna be tough...

<Lots of debugging>

Aha! I found an example where it happens! Seems like it happens before my code.



# Act 2

ME

Hey, when I use the location API for truck #123, it says the truck is in Antarctica. Here's my R code, showing the problem  
<R code>

API DEV

I just checked and I can't reproduce.  
<curl code>

ME

But it reproduces in my code...



# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ ``httrex()`` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:
  - Your R code and output, as in a reprex
  - The HTTP Method & URL (& other info)
  - And also the response from the server in a mostly raw form`

```
library(truckfinder)

(truck <- get_locations(123))
#> --> GET /location?truck_id=123&stops=1 HTTP/1.1
#> --> Host: localhost:8000
#> --> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2
#> --> Accept-Encoding: deflate, gzip
#> --> Accept: application/json, text/xml, application/xml, /*/
#> --
#> <- HTTP/1.1 200 OK
#> <- Date: Sat, 05 Jun 2021 21:30:28 GMT
#> <- Content-Type: application/json
#> <- Content-Length: 167
#> <-
#> << {
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2904,
#> <<       "latitude": -76.6122
#> <<     }
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "latitude": 42.3601,
#> <<       "longitude": -71.0589
#> <<     }
#> <<   ],
#> <<   "destination": [
#> <<     {
#> <<       "latitude": 34.0522,
#> <<       "longitude": -118.2437
#> <<     }
#> <<   ]
#> << }

#> Simple feature collection with 3 features and 1 field
#> Geometry type: POINT
#> Dimension: XY
#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601
#> Geodetic CRS: WGS 84
#> # A tibble: 3 x 2
#>   place           geometry
#>   * <chr>          <POINT [°]>
#> 1 current      (39.2904 -76.6122)
#> 2 origin       (-71.0589 42.3601)
#> 3 destination (-118.2437 34.0522)
```

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ ``httrex()`` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:`

- Your R code and output, as in a reprex

```
library(truckfinder)

(truck <- get_locations(123))
```

```
#> place                      geometry
#> * <chr>                    <POINT [°]>
#> 1 current      (39.2904 -76.6122)
#> 2 origin       (-71.0589 42.3601)
```

```
library(truckfinder)

(truck <- get_locations(123))
#> --> GET /location?truck_id=123&stops=1 HTTP/1.1
#> --> Host: localhost:8000
#> --> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2
#> --> Accept-Encoding: deflate, gzip
#> --> Accept: application/json, text/xml, application/xml, /*
#> --
#> <- HTTP/1.1 200 OK
#> <- Date: Sat, 05 Jun 2021 21:30:28 GMT
#> <- Content-Type: application/json
#> <- Content-Length: 167
#> <-
#> << {
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2904,
#> <<       "latitude": -76.6122
#> <<     }
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "latitude": 42.3601,
#> <<       "longitude": -71.0589
#> <<     }
#> <<   ],
#> <<   "destination": [
#> <<     {
#> <<       "latitude": 34.0522,
#> <<       "longitude": -118.2437
#> <<     }
#> <<   ]
#> << }

#> Simple feature collection with 3 features and 1 field
#> Geometry type: POINT
#> Dimension: XY
#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601
#> Geodetic CRS: WGS 84
#> # A tibble: 3 × 2
#>   place                      geometry
#> * <chr>                    <POINT [°]>
#> 1 current      (39.2904 -76.6122)
#> 2 origin       (-71.0589 42.3601)
#> 3 destination (-118.2437 34.0522)
```

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ ``httrex()`` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:`

- Your R code and output, as in a reprex

```
library(truckfinder)

(truck <- get_locations(123))
```

```
#> place                      geometry
#> * <chr>                    <POINT [°]>
#> 1 current      (39.2904 -76.6122)
#> 2 origin       (-71.0589 42.3601)
```

```
library(truckfinder)

(truck <- get_locations(123))
#> --> GET /location?truck_id=123&stops=1 HTTP/1.1
#> --> Host: localhost:8000
#> --> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2
#> --> Accept-Encoding: deflate, gzip
#> --> Accept: application/json, text/xml, application/xml, /*
#> --
#> <- HTTP/1.1 200 OK
#> <- Date: Sat, 05 Jun 2021 21:30:28 GMT
#> <- Content-Type: application/json
#> <- Content-Length: 167
#> <-
#> << {
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2904,
#> <<       "latitude": -76.6122
#> <<     }
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "latitude": 42.3601,
#> <<       "longitude": -71.0589
#> <<     }
#> <<   ],
#> <<   "destination": [
#> <<     {
#> <<       "latitude": 34.0522,
#> <<       "longitude": -118.2437
#> <<     }
#> <<   ]
#> << }

#> Simple feature collection with 3 features and 1 field
#> Geometry type: POINT
#> Dimension: XY
#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601
#> Geodetic CRS: WGS 84
#> # A tibble: 3 × 2
#>   place                      geometry
#> * <chr>                    <POINT [°]>
#> 1 current      (39.2904 -76.6122)
#> 2 origin       (-71.0589 42.3601)
#> 3 destination (-118.2437 34.0522)
```

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ ``httrex()`` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:`

- Your R code and output, as in a reprex

```
library(truckfinder)

(truck <- get_locations(123))
```

```
#> place                                geometry
#> * <chr>                               <POINT [°]>
#> 1 current     (39.2904 -76.6122)
#> 2 origin      (-71.0589 42.3601)
```

```
library(truckfinder)

(truck <- get_locations(123))
#> --> GET /location?truck_id=123&stops=1 HTTP/1.1
#> --> Host: localhost:8000
#> --> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2
#> --> Accept-Encoding: deflate, gzip
#> --> Accept: application/json, text/xml, application/xml, /*
#> --
#> <- HTTP/1.1 200 OK
#> <- Date: Sat, 05 Jun 2021 21:30:28 GMT
#> <- Content-Type: application/json
#> <- Content-Length: 167
#> <-
#> << {
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2904,
#> <<       "latitude": -76.6122
#> <<     }
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "latitude": 42.3601,
#> <<       "longitude": -71.0589
#> <<     }
#> <<   ],
#> <<   "destination": [
#> <<     {
#> <<       "latitude": 34.0522,
#> <<       "longitude": -118.2437
#> <<     }
#> <<   ]
#> << }

#> Simple feature collection with 3 features and 1 field
#> Geometry type: POINT
#> Dimension: XY
#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601
#> Geodetic CRS: WGS 84
#> # A tibble: 3 × 2
#>   place          geometry
#>   * <chr>        <POINT [°]>
#> 1 current     (39.2904 -76.6122)
#> 2 origin      (-71.0589 42.3601)
#> 3 destination (-118.2437 34.0522)
```

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ `httrex()` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:

- Your R code and output, as in a reprex
- The HTTP Method & URL (& other info)

```
#> -> GET /location?truck_id=123&stops=1 HTTP/1.1
#> -> Host: localhost:8000
```

```
library(truckfinder)

(truck <- get_locations(123))
#> -> GET /location?truck_id=123&stops=1 HTTP/1.1
#> -> Host: localhost:8000
#> -> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2
#> -> Accept-Encoding: deflate, gzip
#> -> Accept: application/json, text/xml, application/xml, */*
#> -
#> -> HTTP/1.1 200 OK
#> -> Date: Sat, 05 Jun 2021 21:30:28 GMT
#> -> Content-Type: application/json
#> -> Content-Length: 167
#> -
#> << {
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2904,
#> <<       "latitude": -76.6122
#> <<     }
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "longitude": 42.3601,
#> <<       "latitude": -71.0589
#> <<     },
#> <<     {
#> <<       "longitude": 34.0522,
#> <<       "latitude": -118.2437
#> <<   ]
#> << }
```

#> << ]
#> << }
#> Simple feature collection with 3 features and 1 field
#> Geometry type: POINT
#> Dimension: XY
#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601
#> Geodetic CRS: WGS 84
#> # A tibble: 3 x 2
#> place geometry
#> \* <chr> <POINT [°]>
#> 1 current (39.2904 -76.6122)
#> 2 origin (-71.0589 42.3601)
#> 3 destination (-118.2437 34.0522)

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ `httrex()` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:

- Your R code and output, as in a reprex
- The HTTP Method & URL (& other info)

```
#> -> GET /location?truck_id=123&stops=1 HTTP/1.1
#> -> Host: localhost:8000
```

```
library(truckfinder)

(truck <- get_locations(123))
#> -> GET /location?truck_id=123&stops=1 HTTP/1.1
#> -> Host: localhost:8000
#> -> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2
#> -> Accept-Encoding: deflate, gzip
#> -> Accept: application/json, text/xml, application/xml, */*
#> ->
#> -> HTTP/1.1 200 OK
#> -> Date: Sat, 05 Jun 2021 21:30:28 GMT
#> -> Content-Type: application/json
#> -> Content-Length: 167
#> -
#> << {
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2904,
#> <<       "latitude": -76.6122
#> <<     }
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "longitude": 42.3601,
#> <<       "latitude": -71.0589
#> <<     }
#> <<   ],
#> <<   "destination": [
#> <<     {
#> <<       "longitude": 34.0522,
#> <<       "latitude": -118.2437
#> <<     }
#> <<   ]
#> << }
```

#> << ]

#> << }

#> Simple feature collection with 3 features and 1 field

#> Geometry type: POINT

#> Dimension: XY

#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601

#> Geodetic CRS: WGS 84

#> # A tibble: 3 x 2

#> place geometry

#> \* <chr> <POINT [°]>

#> 1 current (39.2904 -76.6122)

#> 2 origin (-71.0589 42.3601)

#> 3 destination (-118.2437 34.0522)

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ `httrex()` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:
  - Your R code and output, as in a reprex
  - The HTTP Method & URL (& other info)

```
#> -> GET /location?truck_id=123&stops=1 HTTP/1.1
#> -> Host: localhost:8000
```

```
library(truckfinder)

(truck <- get_locations(123))
#> -> GET /location?truck_id=123&stops=1 HTTP/1.1
#> -> Host: localhost:8000
#> -> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2
#> -> Accept-Encoding: deflate, gzip
#> -> Accept: application/json, text/xml, application/xml, */*
#> ->
#> -> HTTP/1.1 200 OK
#> -> Date: Sat, 05 Jun 2021 21:30:28 GMT
#> -> Content-Type: application/json
#> -> Content-Length: 167
#> -
#> << {
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2904,
#> <<       "latitude": -76.6122
#> <<     }
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "longitude": 42.3601,
#> <<       "latitude": -71.0589
#> <<     }
#> <<   ],
#> <<   "destination": [
#> <<     {
#> <<       "longitude": 34.0522,
#> <<       "latitude": -118.2437
#> <<     }
#> <<   ]
#> << }
```

#> << ]

#> Simple feature collection with 3 features and 1 field

#> Geometry type: POINT

#> Dimension: XY

#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601

#> Geodetic CRS: WGS 84

#> # A tibble: 3 x 2

#> place geometry

#> \* <chr> <POINT [°]>

#> 1 current (39.2904 -76.6122)

#> 2 origin (-71.0589 42.3601)

#> 3 destination (-118.2437 34.0522)

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ ``httrex()`` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:
  - Your R code and output, as in a reprex
  - The HTTP Method & URL (& other info)
  - And also the response from the server in a mostly raw form`

```
#> <<           "longitude": 39.2904,  
#> <<           "latitude": -76.6122  
#> <<           }  
#> <<       ],  
#> <<     "origin": [  
#> <<           {  
#> <<             "latitude": 42.3601,  
#> <<             "longitude": -71.0589
```

```
library(truckfinder)  
  
(truck <- get_locations(123))  
#> --> GET /location?truck_id=123&stops=1 HTTP/1.1  
#> --> Host: localhost:8000  
#> --> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2  
#> --> Accept-Encoding: deflate, gzip  
#> --> Accept: application/json, text/xml, application/xml, */*  
#>  
#> --> HTTP/1.1 200 OK  
#> --> Date: Sat, 05 Jun 2021 21:30:28 GMT  
#> --> Content-Type: application/json  
#> --> Content-Length: 167  
#>  
#> <<   {  
#> <<     "current": [  
#> <<       {  
#> <<         "longitude": 39.2904,  
#> <<         "latitude": -76.6122  
#> <<       }  
#> <<     ],  
#> <<     "origin": [  
#> <<       {  
#> <<         "latitude": 42.3601,  
#> <<         "longitude": -71.0589  
#> <<       }  
#> <<     ],  
#> <<     "destination": [  
#> <<       {  
#> <<         "latitude": 34.0522,  
#> <<         "longitude": -118.2437  
#> <<       }  
#> <<     ]  
#> <<   }  
#> Simple feature collection with 3 features and 1 field  
#> Geometry type: POINT  
#> Dimension: XY  
#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601  
#> Geodetic CRS: WGS 84  
#> # A tibble: 3 x 2  
#>   place           geometry  
#> * <chr>          <POINT [°]>  
#> 1 current      (39.2904 -76.6122)  
#> 2 origin        (-71.0589 42.3601)  
#> 3 destination (-118.2437 34.0522)
```

# R CODE ALONE IS NOT “MINIMAL” ENOUGH

## Introducing `{httrex}`

- ▶ ``httrex()`` - A “reprex”-like interface that makes it easy to re-run your code in a reproducible environment, and includes:
  - Your R code and output, as in a reprex
  - The HTTP Method & URL (& other info)
  - And also the response from the server in a mostly raw form`

```
#> <<           "longitude": 39.2904,  
#> <<           "latitude": -76.6122  
#> <<           }  
#> <<           ],  
#> <<           "origin": [  
#> <<             {  
#> <<               "latitude": 42.3601,  
#> <<               "longitude": -71.0589
```

```
library(truckfinder)  
  
(truck <- get_locations(123))  
#> --> GET /location?truck_id=123&stops=1 HTTP/1.1  
#> --> Host: localhost:8000  
#> --> User-Agent: libcurl/7.64.1 r-curl/4.3 httr/1.4.2  
#> --> Accept-Encoding: deflate, gzip  
#> --> Accept: application/json, text/xml, application/xml, */*  
#>  
#> --> HTTP/1.1 200 OK  
#> --> Date: Sat, 05 Jun 2021 21:30:28 GMT  
#> --> Content-Type: application/json  
#> --> Content-Length: 167  
#>  
#> <<           "current": [  
#> <<             {  
#> <<               "longitude": 39.2904,  
#> <<               "latitude": -76.6122  
#> <<             }  
#> <<           ],  
#> <<           "origin": [  
#> <<             {  
#> <<               "latitude": 42.3601,  
#> <<               "longitude": -71.0589  
#> <<             }  
#> <<           ],  
#> <<           "destination": [  
#> <<             {  
#> <<               "latitude": 34.0522,  
#> <<               "longitude": -118.2437  
#> <<             }  
#> <<           ]  
#> <<           }  
#>  
#> Simple feature collection with 3 features and 1 field  
#> Geometry type: POINT  
#> Dimension: XY  
#> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601  
#> Geodetic CRS: WGS 84  
#> # A tibble: 3 x 2  
#>   place          geometry  
#> * <chr>          <POINT [°]>  
#> 1 current      (39.2904 -76.6122)  
#> 2 origin       (-71.0589 42.3601)  
#> 3 destination (-118.2437 34.0522)
```

# Act 3

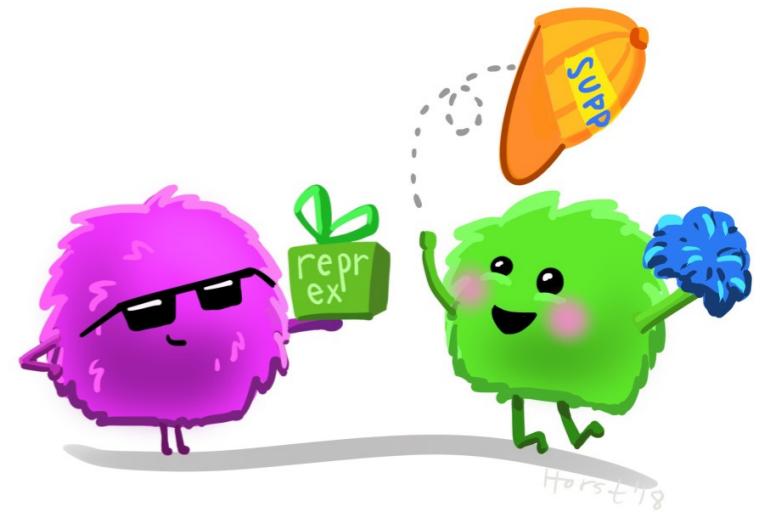
ME

Hm... Oh! it only happens when I ask for the origin and destination locations, too!

<R code & http request info>

API DEV

Oh, you're right, there's a bug where it flips the coordinates when *<complicated reasons why it only happens sometimes involving caching>*. I will fix it, thank you!



# APPENDIX

## Existing solutions

**Don't have time to go into comparing httr to the existing alternatives, but did want to mention them in case you don't know about them (I think `httrex` makes some things easier, but these are often good enough!)**

- ▶ `httr::with\_verbose()` - **Puts output to the console essentially like `httrex`**
- ▶ **Traditional debugging (like breakpoints, traces, etc.)**

# THE END

Thank you to:

- ▶ tidyverse team for `{httr}` and `{reprex}` packages
- ▶ “fuzzy” Illustration by Allison Horst (with some adaptation)
- ▶ YouGov/crunch.io (though views expressed are my own) & previous coworkers/employers
- ▶ useR! organizers and reviewers
- ▶ Julia Evans (@b0rk) writing about debugging (including an upcoming comic)



And thank you for watching!

**Twitter:** [@GregFreedman](#)

**GitHub :** [@gergness](#)

<https://github.com/gergness/httrex>