

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?



Illustration by Allison Horst (<https://github.com/allisonhorst/stats-illustrations>)

Image on right:

Illustration by Allison Horst about reprex, showing a disheveled pink fuzzy ball asking for help from a green fuzzy ball with a hat labelled “support” who looks sad. Then transitions to a much snazzier pink fuzzy ball with a gift wrapped box labelled “reprex” and a much happier support green fuzzy ball.

WHY IS THE TRUCK IN ANTARCTICA?

The motivation for {httrex}

- I was building a shiny app that showed customers where there 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)~~

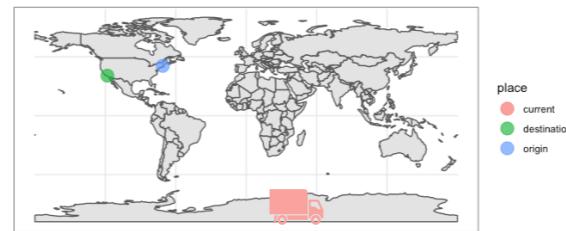


Image on right:

A map showing a pink truck in Antarctica

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.



Illustration adapted from Allison Horst (<https://github.com/allisonhorst/stats-illustrations>)

Image on right:

Illustration adapted from work by Allison Horst showing an angry orange fuzzy ball pointing at a map, and disheveled pink fuzzy ball.

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...



Illustration adapted from Allison Horst (<https://github.com/allisonhorst/stats-illustrations>)

Image on right:

Illustration by Allison Horst about reprex, showing a disheveled pink fuzzy ball asking for help from a green fuzzy ball.

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)

```

Emphasized text:

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

Image on right is of code from htrex output, to be discussed in greater detail in later slides.

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 /locations/123 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 21:38:28 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:  
#> <> Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2904 ymax: 42.3601  
#> <> Geodetic CRS: WGS 84  
#> <> #> 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)
```

Emphasized text:

Your R code and output, as in a reprex

Callouts emphasize parts of image:

```
library(truckfinder)
```

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

and

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

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 /locations?lat=39.2904&lon=-76.6122 HTTP/1.1  
#> Host: localhost:8080  
#> 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, */*  
#> 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  
#>     }  
#>   ]  
#> }  
#> <> 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  
#> <> <list length=3>  
#> <> #> place          geometry  
#> * <chr>          <POINT [...]>  
#> 1 current        (39.2904 -76.6122)  
#> 2 origin         (-71.0589 42.3601)  
#> 3 destination   (-118.2437 34.0522)
```

Callouts further emphasize “123”

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)

```

Callouts further emphasize the longitude and latitude.

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)

```

Emphasized text:

The HTTP Method & URL (& other info)

Callout emphasizes:

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

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/7.84.1 -Curl/4.2.4 libtrix/4.2
#> -> Accept-Encoding: gzip
#> -> Accept: application/json, text/xml, application/xml, */*
#> -> HTTP/1.1 200 OK
#> -> Date: Sat, 05 Jun 2021 21:38:28 GMT
#> -> Content-Type: application/json
#> -> Content-Length: 167
#> ->
#> <<-
#> <<   "current": [
#> <<     {
#> <<       "longitude": 39.2984,
#> <<       "latitude": -76.6122
#> <<     },
#> <<   ],
#> <<   "origin": [
#> <<     {
#> <<       "longitude": -118.2437,
#> <<       "latitude": 34.8522,
#> <<     },
#> <<   ],
#> <<   "destination": [
#> <<     {
#> <<       "longitude": -71.8589,
#> <<       "latitude": 42.3681
#> <<     }
#> <<   ]
#> << }
#> << Simple feature collection with 3 features and 1 field
#> << Geometry type: POINT
#> << Dimension: XY
#> << Bounding box: xmin: -118.2437 ymin: -76.6122 xmax: 39.2984 ymax: 42.3681
#> << Geodetic CRS: WGS 84
#> << #> A matrix of 3 x 2
#> << place           geometry
#> << * <chr>           <POINT (...)>
#> << 1 current      (39.2984 -76.6122)
#> << 2 origin       (-118.2437 34.8522)
#> << 3 destination (-71.8589 42.3681)
```

Callout further emphasizes “truck_id=123”

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)

```

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_truck_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, */*  
#> <- Content-Type: application/x-www-form-urlencoded  
#> <- Content-Length: 167  
#> <- Date: Sat, 05 Jun 2021 21:38:28 GMT  
#> <- HTTP/1.1 200 OK  
#> <- {  
#> <-   "current": [  
#> <-     {  
#> <-       "longitude": 39.2904,  
#> <-       "latitude": -76.6122  
#> <-     }  
#> <-   ],  
#> <-   "origin": [  
#> <-     {  
#> <-       "latitude": 42.3601,  
#> <-       "longitude": -71.0589  
#> <-     }  
#> <-   ],  
#> <-   "destination": [  
#> <-     {  
#> <-       "latitude": 34.8522,  
#> <-       "longitude": -118.2437  
#> <-     }  
#> <-   ]  
#> <- }  
#> <- Single 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  
#> <- As table: 3 x 2  
#> <- place geometry  
#> <- <chr> <POINT (...)>  
#> 1 current (39.2904 -76.6122)  
#> 2 origin (-71.0589 42.3601)  
#> 3 destination (-118.2437 34.8522)
```

Emphasized text:

And also the response from the server in a mostly raw form

Callout emphasizes:

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

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_truck_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, */*  
#> <- Content-Type: application/x-www-form-urlencoded  
#> <- Content-Length: 167  
#> <- Date: Sat, 05 Jun 2021 21:38:28 GMT  
#> <- Connection: keep-alive  
#> <-  
#> << {  
#> <<   "current": [  
#> <<     {  
#> <<       "longitude": 39.2904,  
#> <<       "latitude": -76.6122  
#> <<     },  
#> <<   ],  
#> <<   "origin": [  
#> <<     {  
#> <<       "latitude": 42.3601,  
#> <<       "longitude": -71.0589  
#> <<     }  
#> <<   ]  
#> <<   "destination": [  
#> <<     {  
#> <<       "latitude": 34.8522,  
#> <<       "longitude": -118.2437  
#> <<     }  
#> <<   ]  
#> <<  
#> <- Single 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  
#> <- Features: 3 x 2  
#> <- place geometry  
#> * <chr> <POINT (...)>  
#> 1 current (39.2904 -76.6122)  
#> 2 origin (-71.0589 42.3601)  
#> 3 destination (-118.2437 34.8522)
```

Callout further emphasizes ““longitude”: 39.2904” & ““latitude”: 42.3601,”

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!

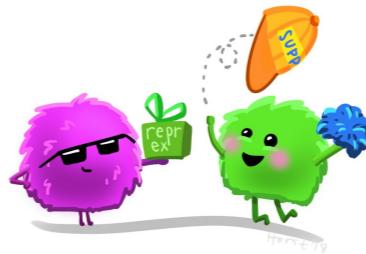


Illustration adapted from Allison Horst (<https://github.com/allisonhorst/stats-illustrations>)

Image on right, snazzier pink fuzzy ball with a gift wrapped box labelled “reprex” and a much happier support green fuzzy ball

APPENDIX

Existing solutions

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

- ``httr::with_verbose()`` - Puts output to the console essentially like ``httrrex``
- 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>

Image on right, httr package logo with a T-Rex running from a comet.