



# Building Web APIs in R with Plumber

*Jeff Allen*  
*2018-07-25*  
*RStudio*

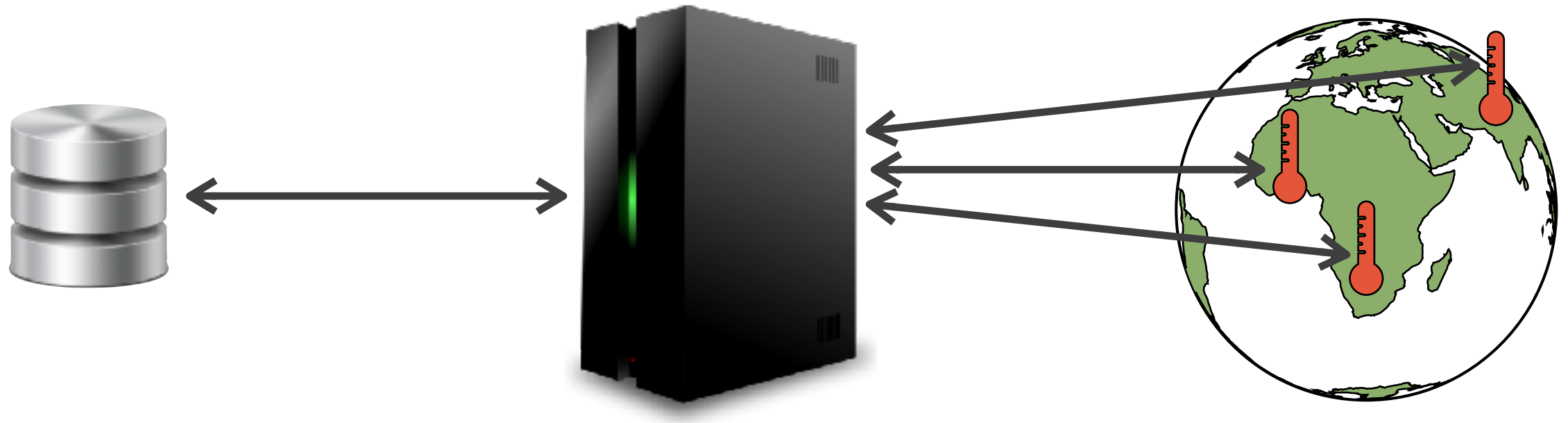
1. What is an API?
2. What is Plumber?
3. Demo
4. Tips & tricks

1. **What is an API?**
2. What is Plumber?
3. Demo
4. Tips & tricks

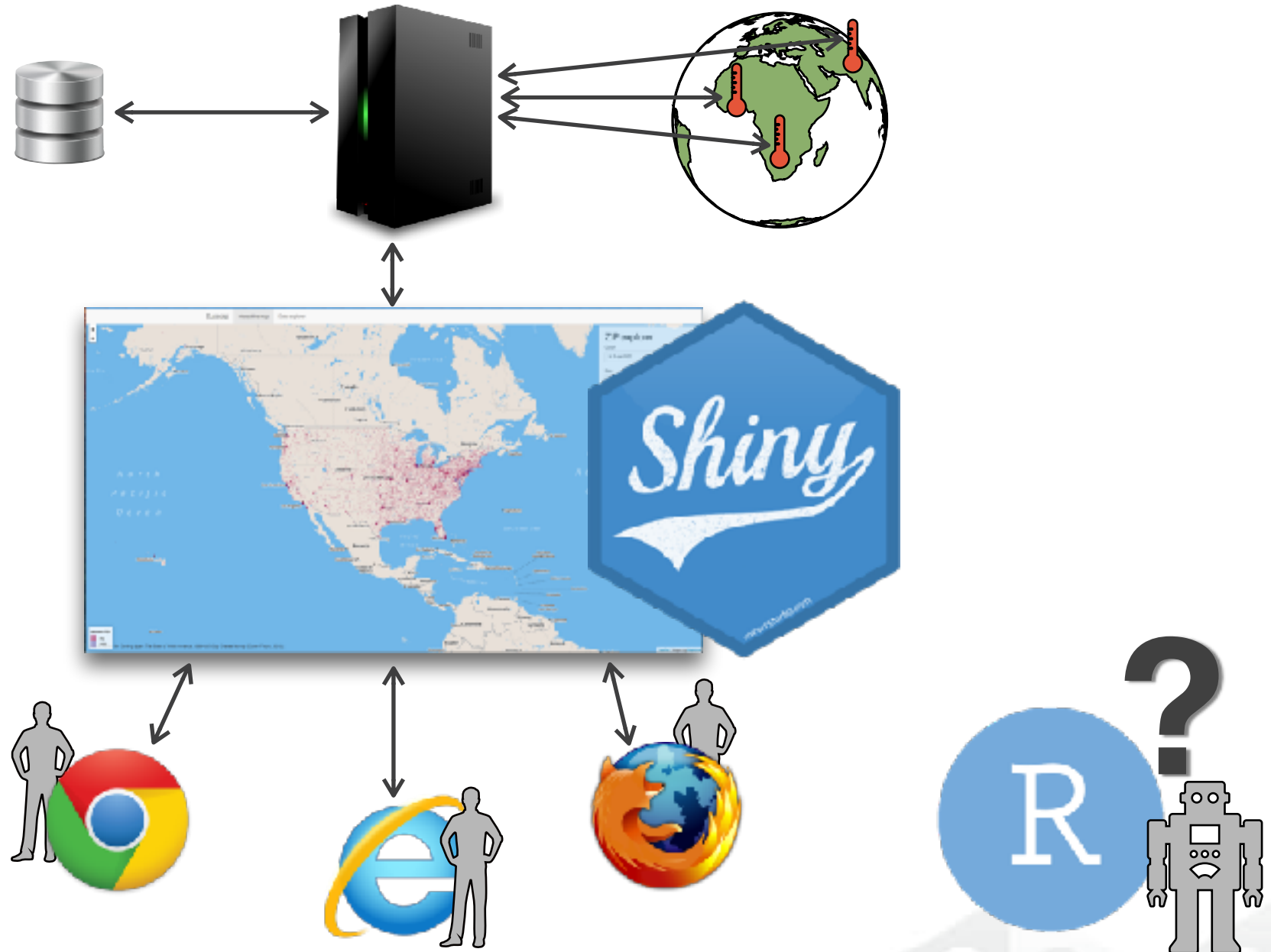
# Web APIs

- Machine-friendly data formats (JSON)
- Programmatically consumed
- Accessible from all major programming languages and platforms
  - Including R (httr), and web browsers

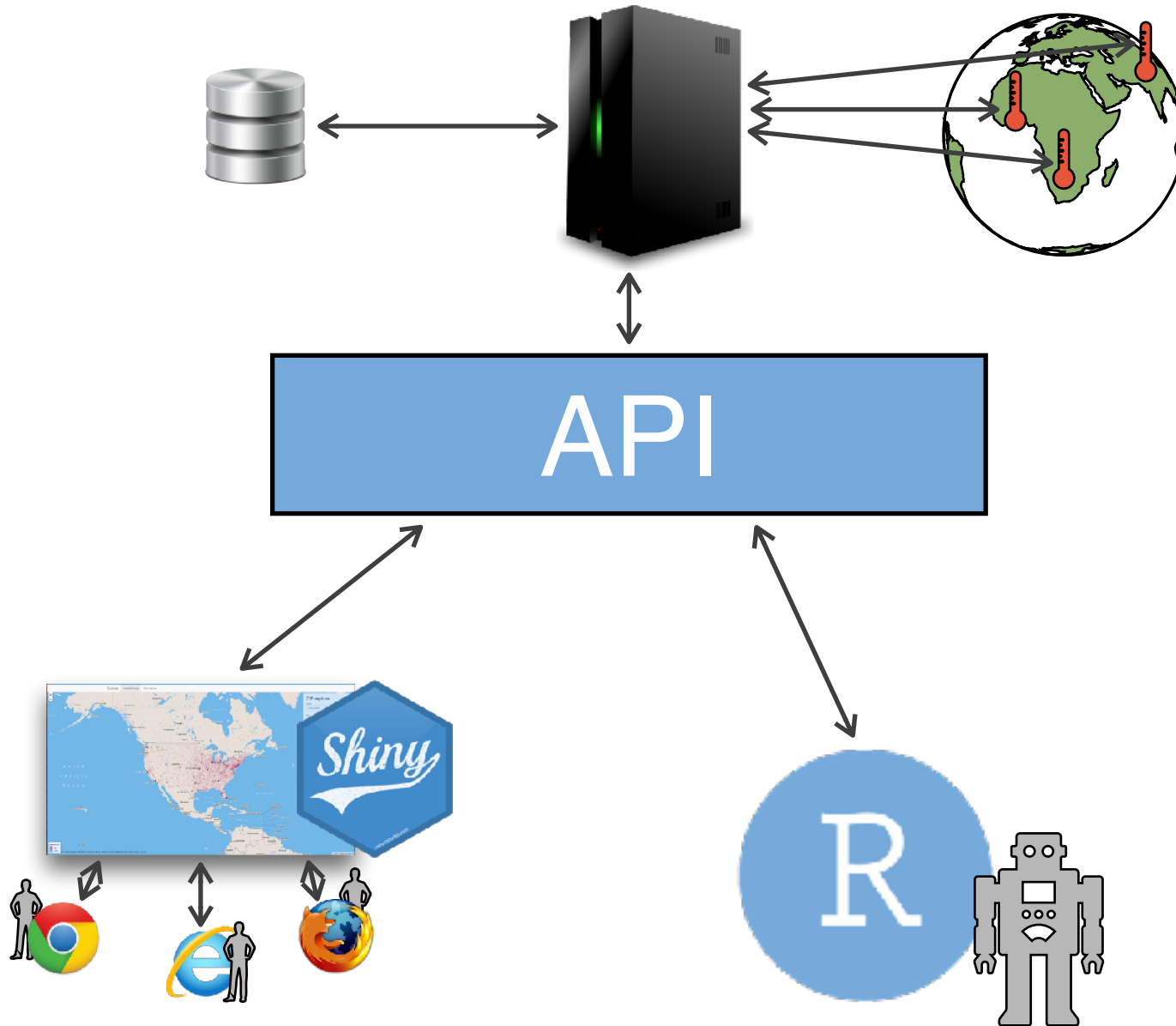
# Weather System



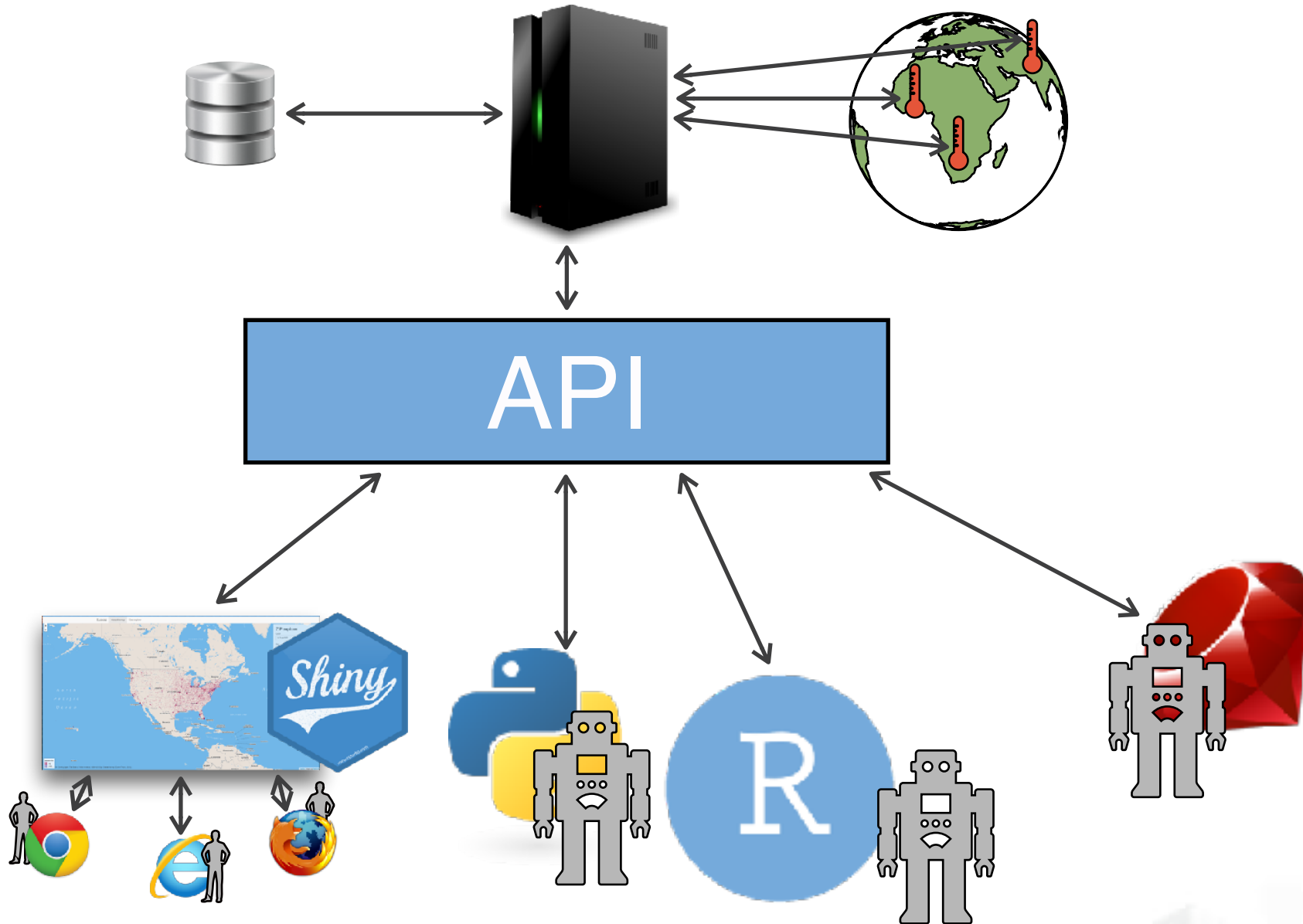
# Weather System



# Weather System



# Weather System





# Hypertext Transfer Protocol (HTTP)

- Specify the “path” of the request: `/hello.html`
- Different types of requests: GET, POST, PUT, DELETE, etc.
  - Browsers default to GET
- “REST” is a common convention for organizing APIs

plumber Docs Examples - See code on GitHub

user integration. You might want to have R respond to new issues created for a repository, evaluate changes to data or code on GitHub, or just about anything else you can imagine.

## Setup GitHub Webhook

The best guide to setting up a GitHub Webhook is the [official documentation page](#). In brief...

Inspector Console Debugger {} Style Editor Network >>

Filter URLs || ☐ Persist Logs ☐ Disable cache No throttling HAR

All HTML CSS JS XHR Fonts Images Media WS Other

Status	Method	F	Headers	Cookies	Params	Response	Timings	Security
200	GET	github	Request URL: https://www.rplumber.io/examples/github					
200	GET	main	Request method: GET					
200	GET	4276	Remote address: 206.81.1.198:443					
200	GET	jquery	Status code: 200 ⓘ Edit and Resend Raw headers					
200	GET	f17b	Version: HTTP/2.0					
200	GET	emb	Filter headers					
200	GET	vers	Response headers (360 B)					
			Request headers (396 B)					
			Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8					
			Accept-Encoding: gzip, deflate, br					
			Accept-Language: en-US,en;q=0.5					

7 requests 358.09 KB

1. What is an API?
- 2. What is Plumber?**
3. Demo
4. Tips & tricks

# APIs with Plumber

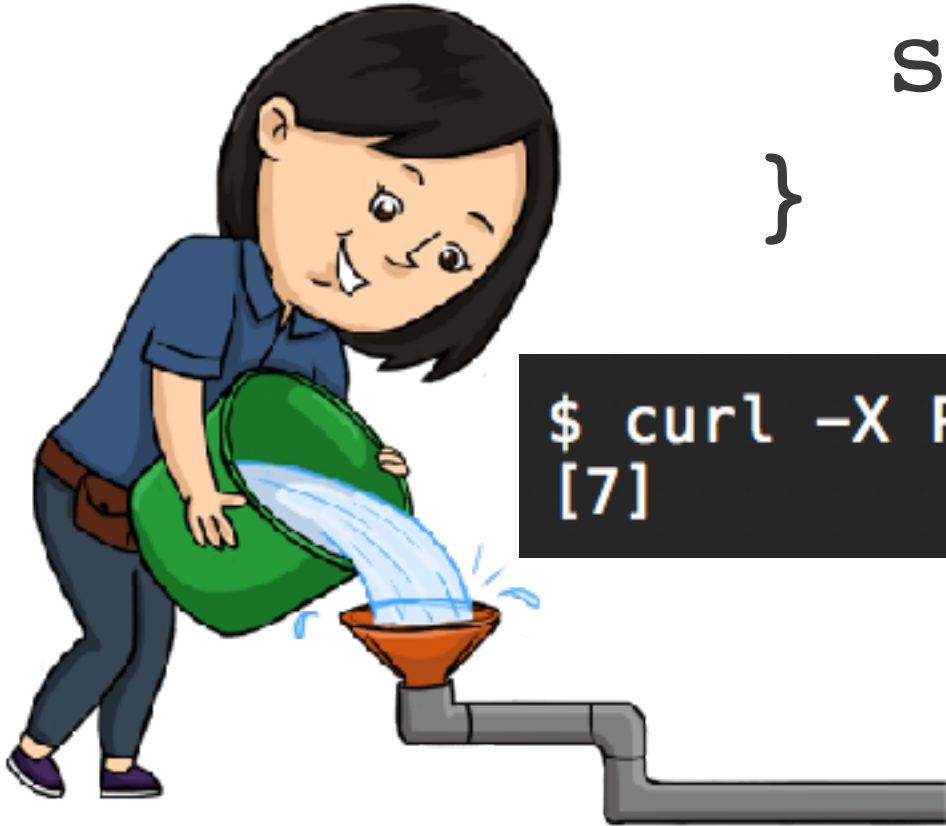
- Open-source R package
- Converts your existing R code into web APIs using special comments
- `install.packages("plumber")`



```
## @get /echo  
function(msg) {  
  paste("You said:", msg)  
}
```



```
## @post /sum  
function(a, b) {  
  sum(a, b)  
}
```



```
$ curl -X POST "http://localhost:8000/sum?a=3&b=4"  
[7]
```

library(**dygraphs**)

```
#' @get /spots/<year>/graph
```

```
#' @serialize
```

```
function(year)
```

```
  dygraph(data)
```

```
    main =
```

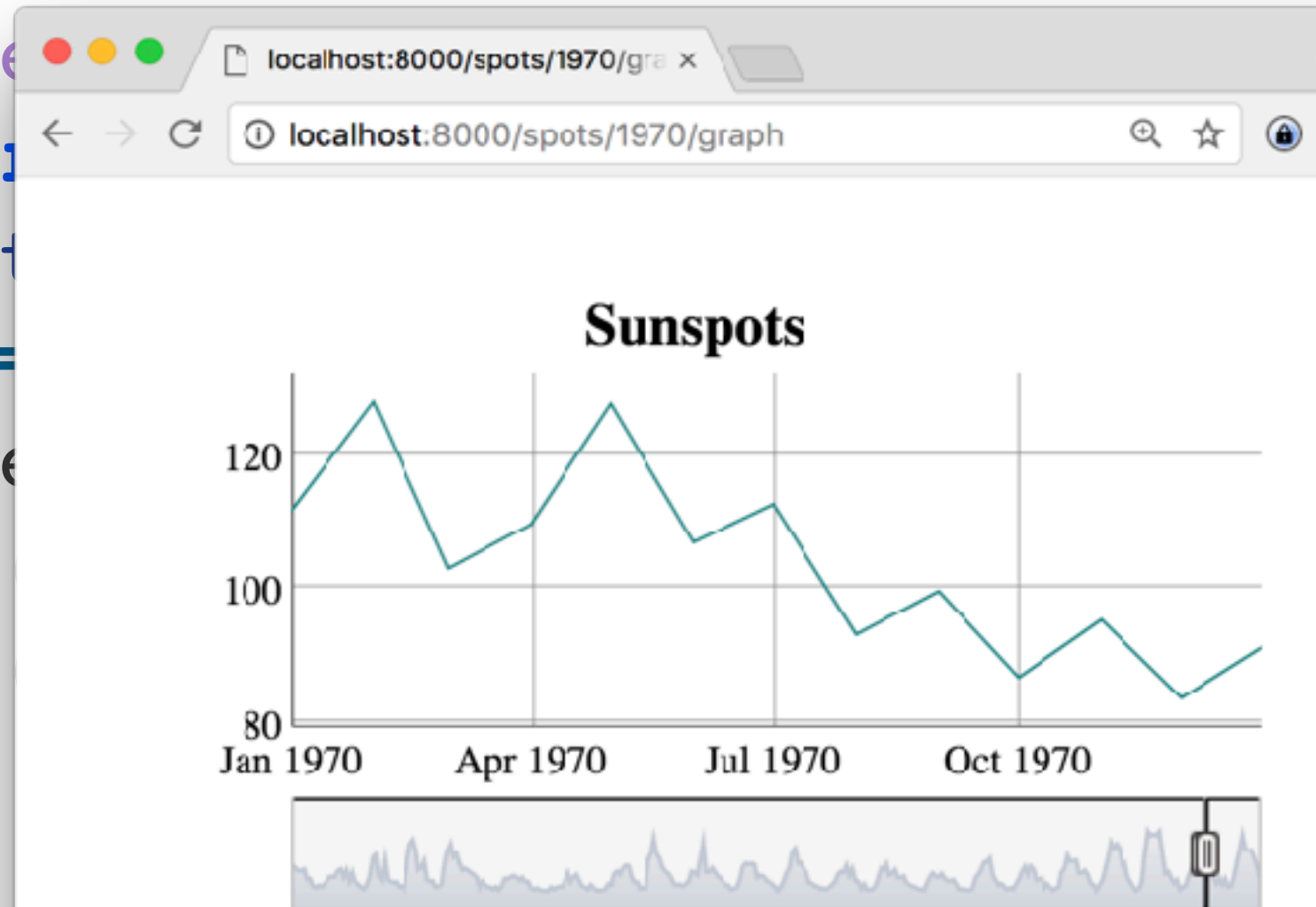
```
    dyRangeSe
```

```
    paste0
```

```
    paste0
```

```
)
```

R Studio }



# APIs with Plumber

```
library(plumber)  
pr <- plumb("plumber.R")  
pr$run(port=8000)
```

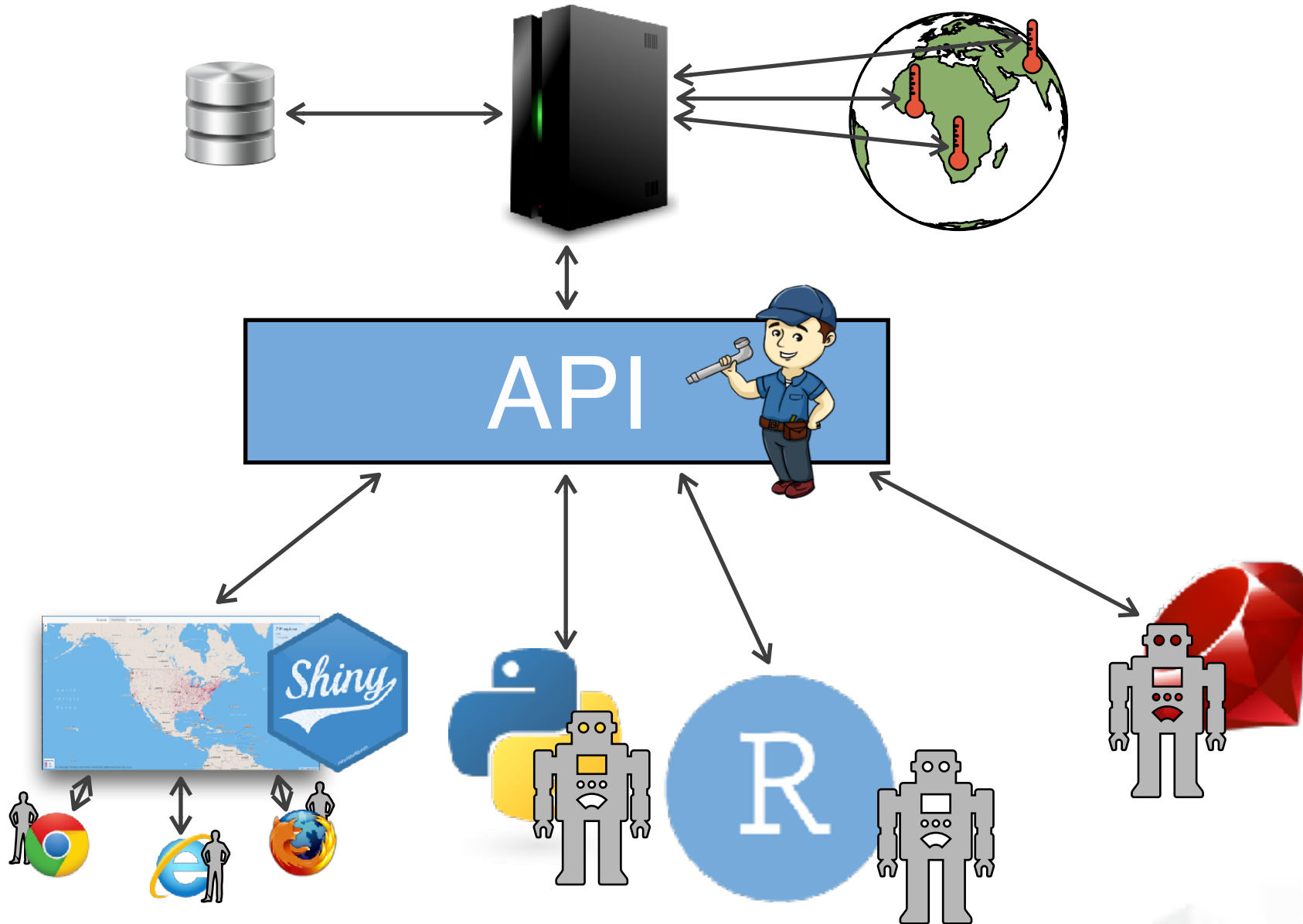
# OR

```
plumber::plumb("plumber.R")$run(port=8000)
```





# Weather System



# Compare to...

- Porting your existing software to R
- Porting your existing R code to another language

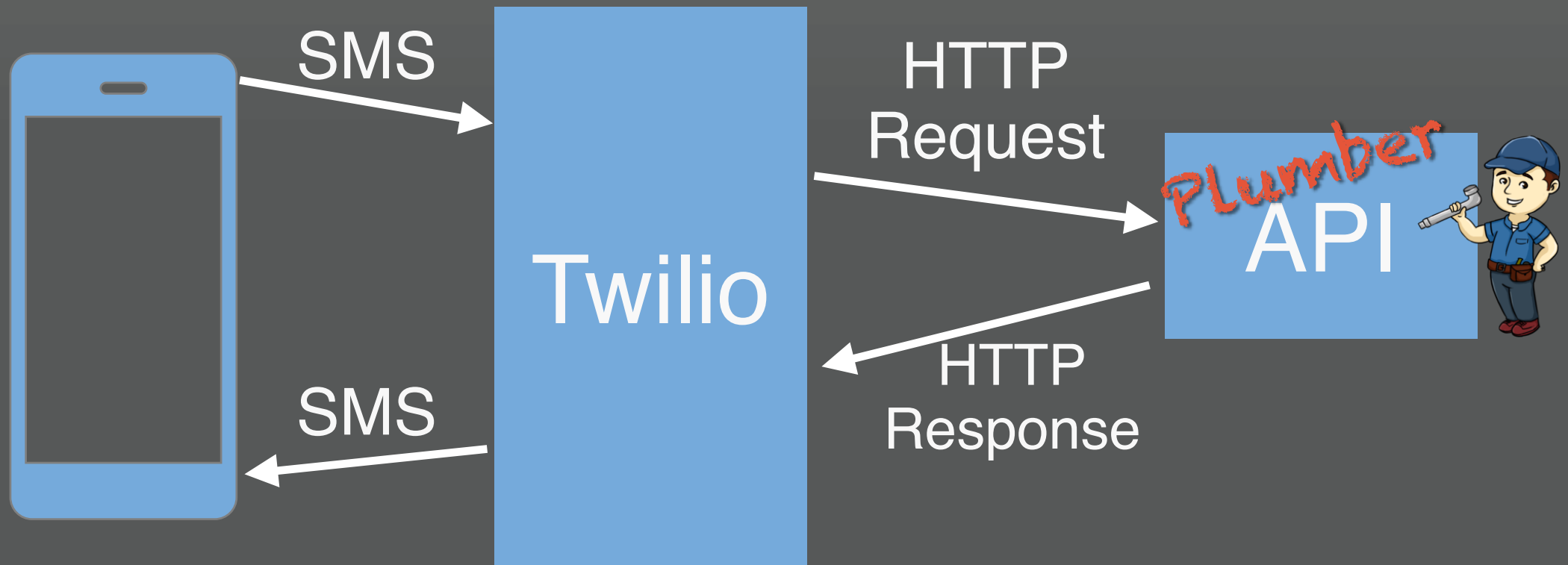


1. What is an API?
2. What is Plumber?
3. **Demo**
4. Tips & tricks

Text recently used emojis to  
972-218-0013



Text recently used emojis to  
972-218-0013



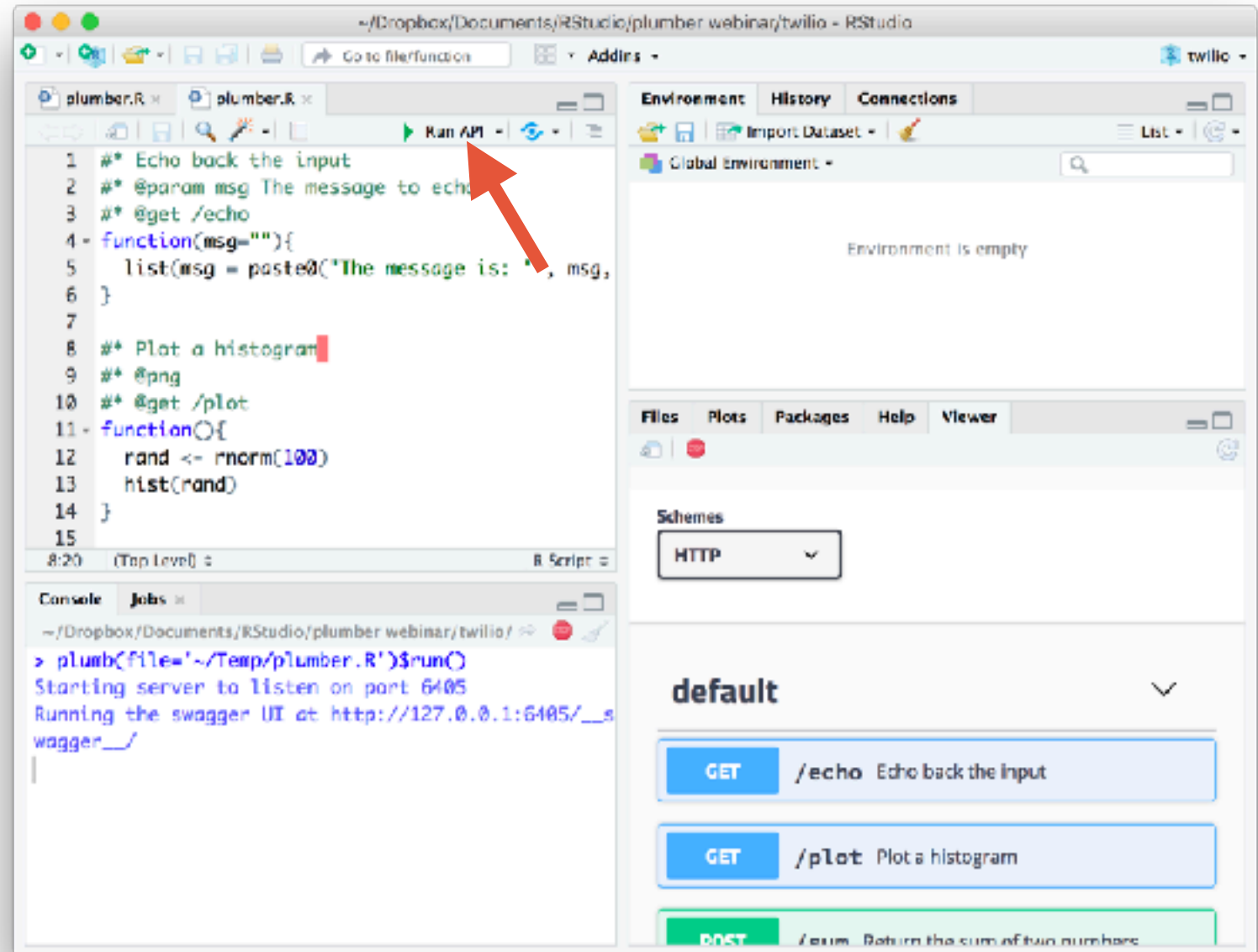
1. What is an API?
2. What is Plumber?
3. Demo
- 4. Tips & tricks**

# Hosting

- RStudio Connect
  - `rsconnect::deployAPI()`
  - Load balancing, access controls
  - <https://rstd.io/rsc>
- Digital Ocean
  - `do_provision()`

# IDE Integration

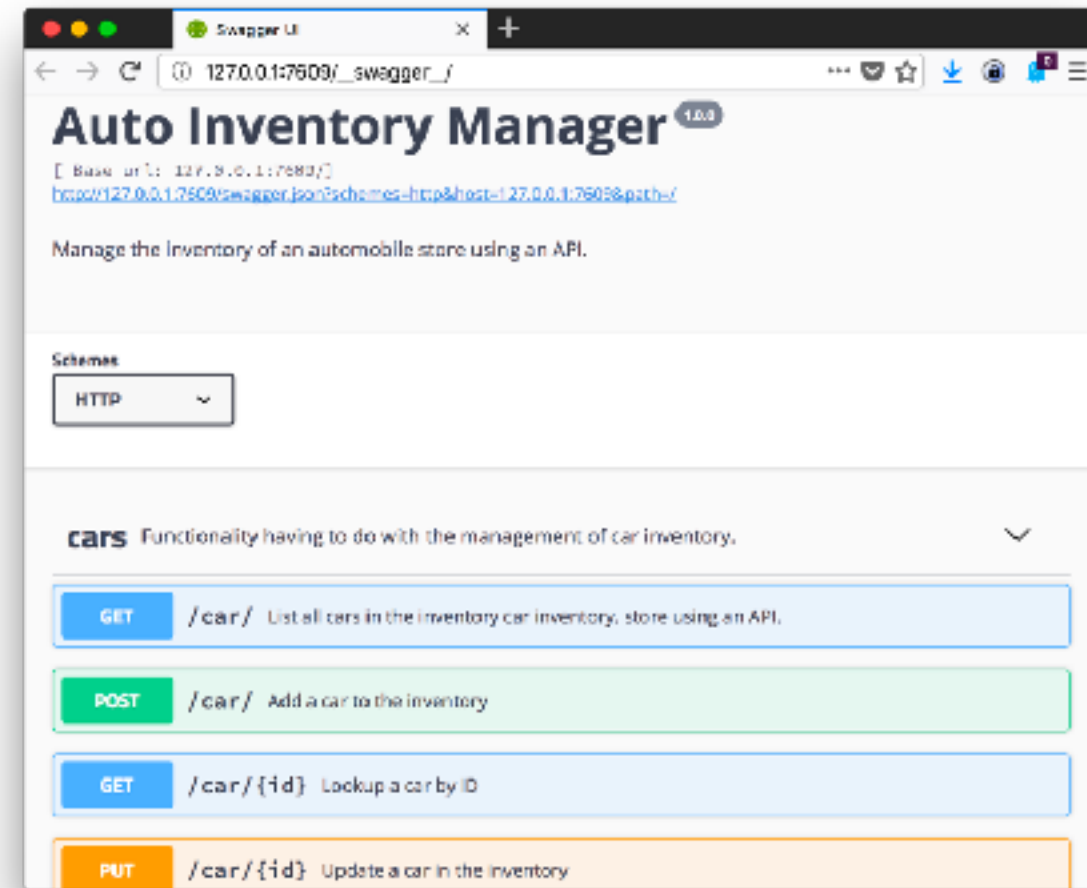
- Available in 1.2 preview
- Brings up Swagger automatically





# Swagger

- Support the OpenAPI specification with accompanying tooling
- Auto-generated, control with  
`$run (swagger=TRUE)`



# Debugging

- Pause and inspect
- Mutate values
- Continue

The screenshot shows the RStudio interface with a Plumber API endpoint being debugged. The console shows the API call, the environment shows variables 'a' and 'b', and the traceback shows the function call.

```
10 #* @get /plot
11 function(){
12   rand <- rnorm(100)
13   hist(rand)
14 }
15
16 #* Return the sum of two numbers
17 #* @param a The first number to add
18 #* @param b The second number to add
19 #* @post /sum
20 function(a, b){
21   browser()
22   as.numeric(a) + as.numeric(b)
23 }
```

Environment

Variable	Value
a	"4"
b	"3"

Traceback

```
(function (a, b) at plumber.R:21)
(function (...)
```

Console

```
16 11 (Top level)
Browse[1]> a
[1] "4"
Browse[1]> b
[1] "3"
Browse[1]> |
```

Response

Response content type: application/json

Curl

```
curl -X POST "http://127.0.0.1:7824/sum?b=3&a=4" -H "accept: application/json"
```

```
#' @get /count
function( ) {
  read_csv_async( "file.csv" ) %...>%
  nrow( )
}
```

## Upcoming

- Async support
- Testing?
- Expanded documentation
- Additional maintainers

# Questions?

- <https://rplumber.io>
- StackOverflow: `plumber` tag
- GitHub: `trestletech/plumber`
- Slides: <http://rstd.io/plumber-webinar>