GET and POST requests in theory

WORKING WITH WEB DATA IN R

Oliver Keyes
Instructor

HTTP requests

- Conversation between your machine and the server
- First: what you want to happen
- "methods" - different requests for different tasks

GET and POST

- GET: "get me something"
- POST: "have something of mine"

Other types

- HEAD - just like `head()`
- DELETE - "remove this thing"
- Many others! But GET and POST are the big ones

Making GET requests with httr

```
response <- GET(url = "https://httpbin.org/get")
content(response)
```

```
$args
named list()

$headers
$headers$Accept
[1] "application/json, text/xml, application/xml, */*"
...
```

Making POST requests with httr

```
response <- POST(url = "https://httpbin.org/post")
```

Let's practice!

WORKING WITH WEB DATA IN R

Graceful httr

WORKING WITH WEB DATA IN R

Charlotte Wickham
Instructor

Error handling

Every response includes an HTTP **status code**.

Error handling

```
response <- GET("https://httpbin.org/get")
response
```

```
Response [https://httpbin.org/get]
  Date: 2017-08-24 20:29
  Status: 200
  Content-Type: application/json
  Size: 330 B
{ ...
```

Understanding status codes

- Code starts with:
  - 2 - great!
  - 3 - great!
  - 4 - your code is broken
  - 5 - their code is broken
- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- Check for bad codes with `http_error()`

URL construction

- Most of URL doesn't change
- Stitch URLs together from bits that don't change with the bits that do
- Saves thinking and typing

Directory-based URLs

- Slash-separated, like directories
- `https://fakeurl.com/api/peaches/thursday`
- Use `paste()` , with `sep = "/"`

Parameter-based URLs

- Uses URL parameters (`a=1&b=2`)
- `https://fakeurl.com/api.php?fruit=peaches&day=thursday`
- Use `GET()` to construct the URL with `query` argument

Let's practice!

WORKING WITH WEB DATA IN R

Respectful API Usage

WORKING WITH WEB DATA IN R

Oliver Keyes
Instructor

User agents

- Bits of text that ID your browser (or software)
- Gives the server some idea of what you're trying to do
- You can set one with your requests with `user_agent()`
- Add an email address so they can contact you.

Rate limiting

- Too many requests makes for a sad server
- Deliberately slows down your code to keep under a desired "rate"
- Slows you, but avoids getting you banned from the server

Let's practice!

WORKING WITH WEB DATA IN R