API'S

Shiny from RStudio™

OUTLINE

- Course review
- What are API's and what do they do?
- Making an API Call
  1. Build your URL
  2. Encode the URL
  3. Process the content
  - Spatial Data with Esri
- Geocode Example
- Shiny Example

Survey Monkey

Whats an API,

and what does it do?

API EXAMPLES

- WPRDC
- Census
- Geocoders
- Esri Online Datasets
- Online Weather APIs
- Sport Score API
- And more!

API'S

- Stands for: Application Programming Interface

- There are many kinds of API's
  - *Web service*
    - SOAP, XML-RPC, JSON-RPC, and **REST**
  - *WebSocket*
  - *Library-based*
  - *Class-based*
  - *OS functions and routines*
  - *Object remoting*
  - *Hardware*

REST API'S

- End points - different URL's that tell the webserver what data you would like
- It's essentially a website where you request different "end points"
- There are 5 types of Requests you can make
  - `GET` (what we will use the most in this course)
  - `POST` *(sometimes necessary for authentication, if you're trying to write data somewhere)*
  - `PUT`
  - `PATCH`
  - `DELETE`

Making an API Call

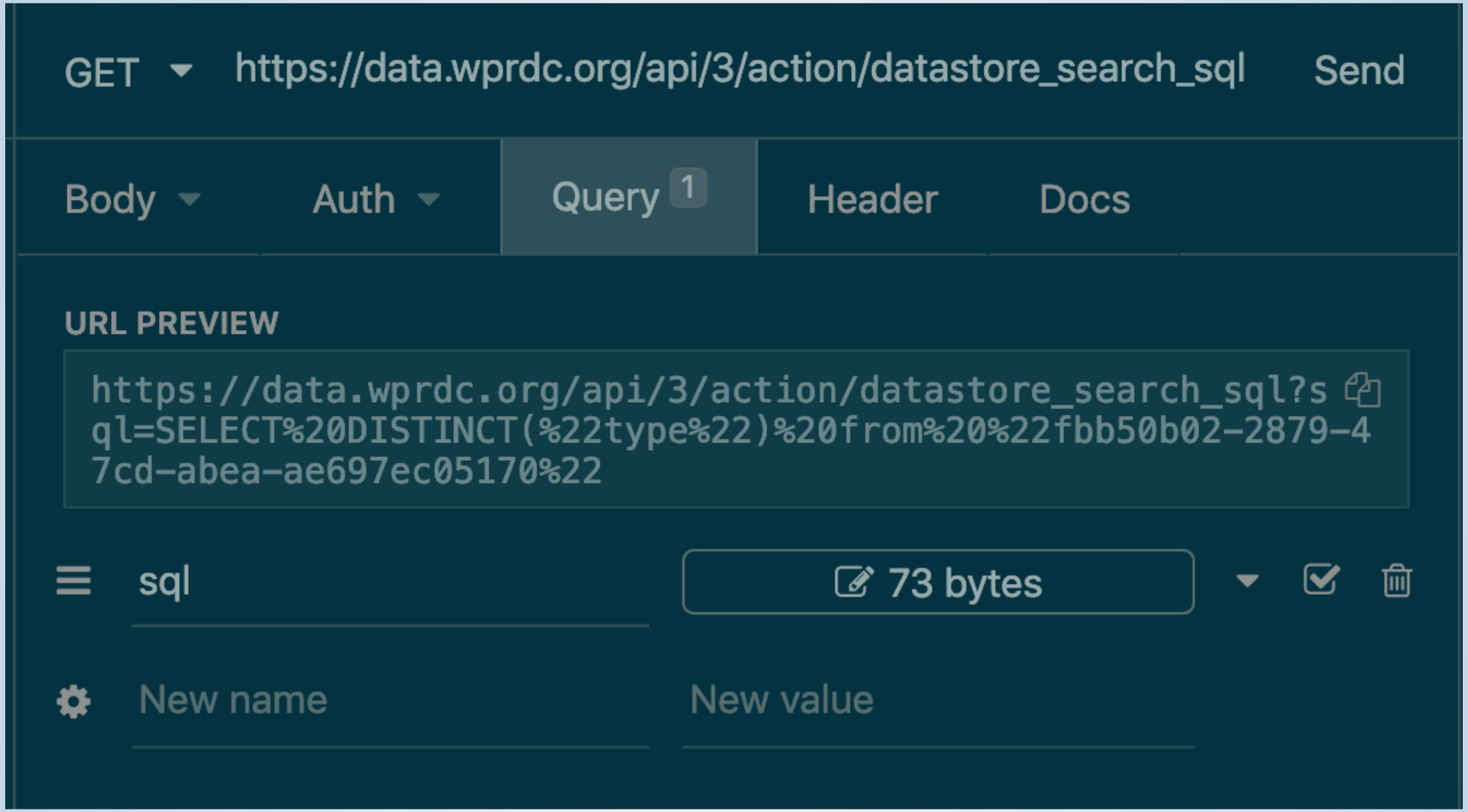
THE STEPS

1. Build your URL
2. Encode the URL
3. Process the content
4. Transform to a usable format

1. BUILDING YOUR QUERY

Many tools that make life easier:

- *Insomnia*
- *Advanced REST Client*
- *PostMan*
- *And others…*



WPRDC API Call in Insomnia

THE STEPS

1. Build your URL
2. Encode the URL
3. Process the content
4. Transform to a usable format

DEMO

```
URLencode("someString", repeated = TRUE)
```

THE STEPS

1. Build your URL
2. Encode the URL
3. Process the content
4. Transform to a usable format

CONTENT

- Any API call will have multiple portions of it.
- 2 most important are:
  - Content
  - status_code

GETTING TO THE CONTENT

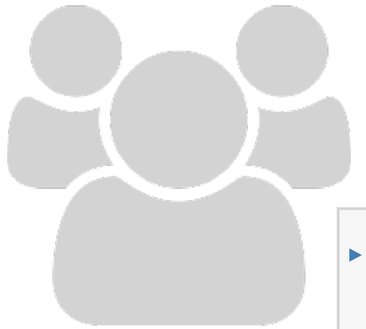
- Most API calls you will be making are GET requests.

```
get <- httr::GET("encodedURL")
c <- jsonlite::fromJSON(content(get, "text"))
```

- Arguments you may need:
  - `$something` after `fromJSON` function
  - `flatten=TRUE`

ERRORS

- Status codes indicate the result of the HTTP request.
  - ***100's*** - *info*
  - ***200's*** - *success*
  - ***300's*** - *redirection*
  - ***400's*** - *client error (you messed up)*
  - ***500*** *'s- server error (something went wrong on their end, but you still could have messed up)*

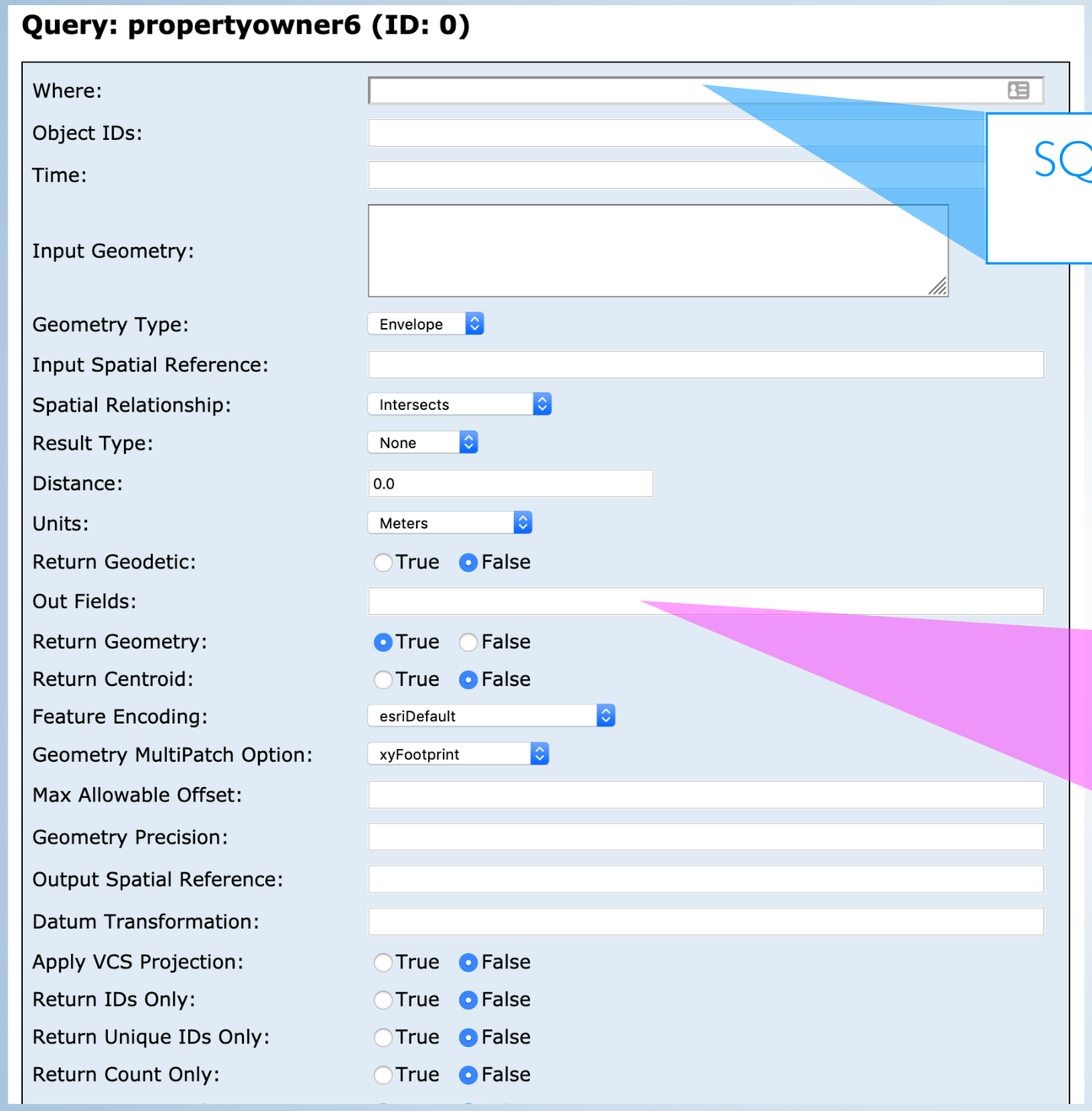


EXERCISE

- Open exercises/api_practice.Rmd and use the chunk labeled “Blotter”
  - Like last class generate an API call that downloads all of the data from the City of Pittsburgh Police Blotter
    - It might be easier to build the query in Insomnia or something else first
    - Stretch: After you have built a query that calls all of the data, add a group by or filter of some kind

10m 00s

Spatial Data



SQL like where statement to get one the data you want

Same as the select portion of a SQL query

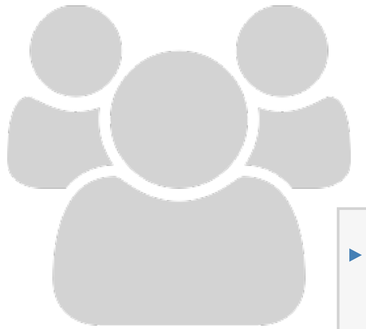
Source: Allegheny County Esri API

GETTING SPATIAL DATA

- For ESRI API's so long as your format is set to GEOJSON…

```
data <- readOGR("encodedURL")
```

- Its that easy



EXERCISE

- Open exercises/api_practice.Rmd and go to the chunk labeled “Esri”
  - Look at the fields on the May 2019 Election layer from the Allegheny County Esri API: https://services1.arcgis.com/vdNDkVykv9vEWFX4/ArcGIS/rest/services/Allegheny_County_Polling_Places_May2019/FeatureServer/0
    - Get all of the polling places in just the City of Pittsburgh and load it into R from the URL

10m 00s

SOLUTION

Solutions to both of todays exercises are in:
api_practice_solutions.R

THE STEPS

1. Build your URL
2. Encode the URL
3. Process the content
4. Transform to a usable format

Other API’s

OTHER API'S

- Not all (most) API's will require you to do filtering or sql in them
  - This is mostly how Data Portal's API's work
  - Socrata Portals have a weird endpoint sql hybrid model using their SoQL framework: https://dev.socrata.com/docs/queries/
- Typically API's will have endpoints
- You will need to read up on the Documentation on the API you are attempting to use.

Geocode Example

DEMO

alco_geocode.R

Shiny

Example

DEMO

`app/311_dashboard.R`

API'S

Shiny from RStudio™


That's how extra credit
is supposed to feel.