APIs

INFO 201

Today’s Objectives

Practice debugging functions

Learn how to ***read data*** from a web REST API

Learn how to ***format data*** that comes in JavaScript Object Notation (JSON)

Debugging functions

What do you do when you get an error in a function that you're writing?

Debugging Functions

Read error messages

Google, look at documentation

Set your parameters to sample values, and run through your function line-by-line

module 9 exercise-4

Application Programming Interfaces

APIs

Tools, protocols for software development

Exposes components (data, functions) in documented formats

Have we used any APIs?

An example:

```
# install.packages('ggmap')
library(ggmap)
qmap('Seattle')
```

REST APIs

Representational State Transfer APIs

Exposes **data** components

Transfers information with HTTP (HyperText Transfer Protocol)

Enables querying content over the web

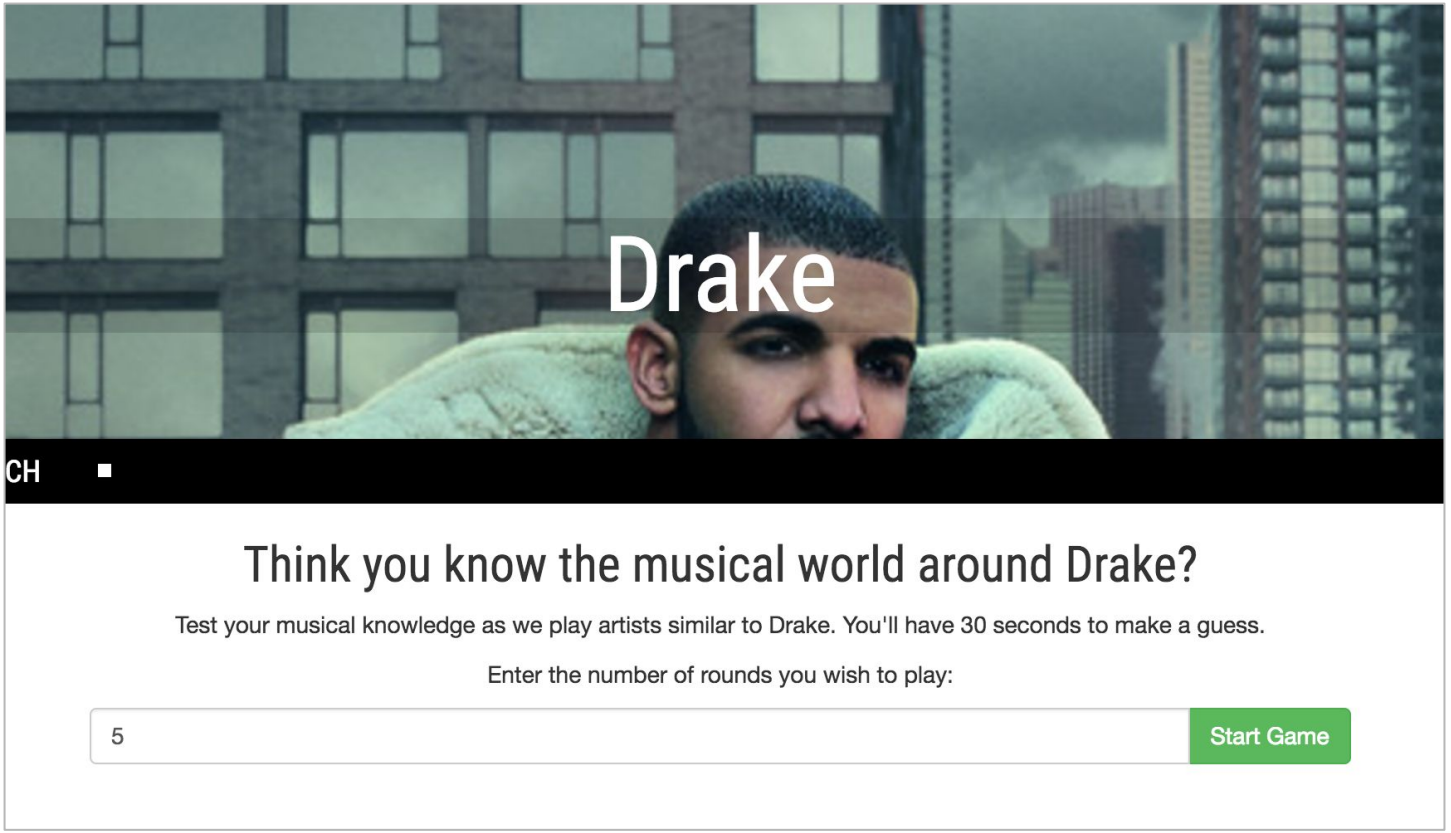
How it works

Navigate to a URL that specifies the data you want

```
https://api.spotify.com/v1/search?q=adele&type=artist
```

Returns information in JavaScript Object Notation (list of lists)

```
{
    artists: {
        href: "https://api.spotify.com/v1/searc...",
        items: [{
            external_urls: {
            spotify: "https://open.spotify...."
        },
        followers: {
            href: null,
            total: 4093432
        },
    }
}
```



Inspiration

REST APIs in R

Construct a search

```
# Base URL of API
base <- 'https://api.spotify.com/v1/search?'

# Parameters
search <- 'q=adele'
type <- '&type=artist'

# Query string
query.url <- paste0(base, search, type)
```

Use the fromJSON function (jsonlite package)

```
data <- fromJSON(query.url)
```

module 10 exercise-2

Formatting Data

Sometimes, the data is in a strange format...

```
# Let's do something silly
people <- data.frame(names = c('Spencer', 'Jessica', 'Keagan'))
favorites <- data.frame(
                food = c('Pizza', 'Pasta', 'salad'),
                music = c('Bluegrass', 'Indie', 'Electronic')
             )
# Store dataframe column
people$favorites <- favorites
```

Use flatten function (from jsonlite) to create separate columns

```
# Spread a dataframe into separate columns
people <- flatten(people)
```

module 10 exercise-3

Upcoming...

By Thursday: Be confident with **module 10**

Due Tuesday, 11/1 (***before class***): a5-github-report