

Introduction to APIs and JSON

Credits: go.ncsu.edu/introapis

Outline

- Intro to APIs
- Intro to JSON
- Exercise: Explore an API

Intro to APIs

What is an API?

- Application Programming Interface
- Building blocks for developing a computer program
- Web-based system, operating system, software library, etc
- A set of rules and procedures that facilitate interactions between computers and their applications

Note: examples include Windows API for working with Windows OS and Android API for their development kit

See: https://en.wikipedia.org/wiki/Application_programming_interface

Web APIs

When used in the context of web development, an API is typically a defined set of specifications, such as [Hypertext Transfer Protocol](#) (HTTP) request messages, along with a definition of the structure of response messages, which is usually in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format.

Web APIs

- allows users to query a remote database over the internet
- take on a variety of formats
- majority adhere to a particular style known as **Representational State Transfer** or **REST**
- "RESTful" APIs are convenient because we can use them to query databases using URLs



The API

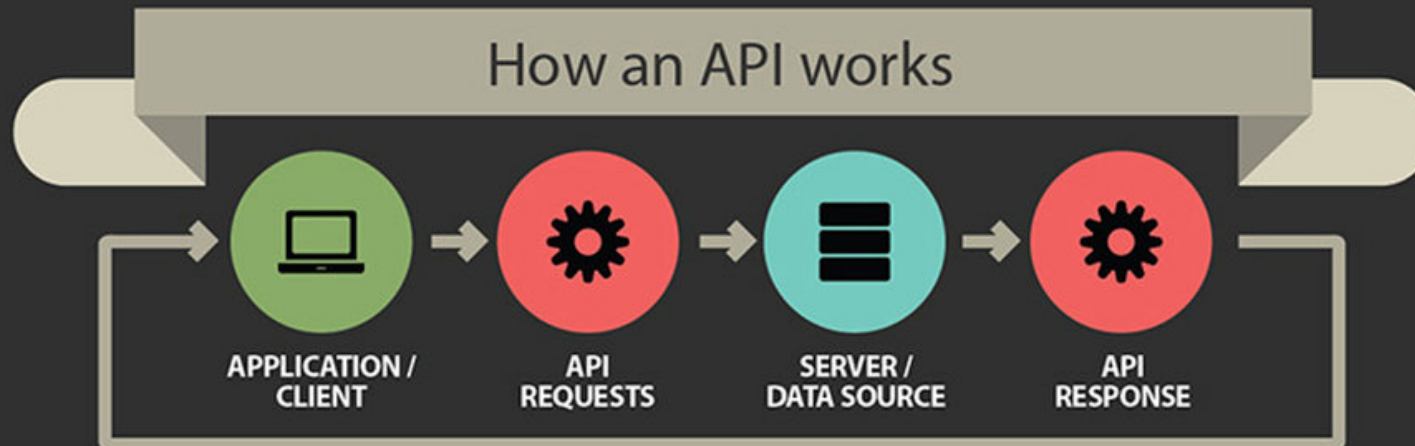
Application Programming Interface



API Definition

An application program interface that provides a developer with programmatic access to a proprietary software application. A software intermediary that makes it possible for application programs to interact with each other and share data.¹

How an API works



Retrieved from [What's an API and Why Do You Need One?](#)

RESTful Web APIs are All Around You...

Consider a simple Google search.

Go ahead and search something.

Ever wonder what all that extra stuff in the address bar was all about?

RESTful Web APIs are All Around You...

It looks like Google makes its query by taking the search terms, separating each of them with a " + ", and appending them to the link:

```
https://www.google.com/#q=
```

So that we have

```
https://www.google.com/#q=search1+search2
```

So can change our Google search by adding some terms to the URL.

Some Basic Terminology: URL

- Uniform Resource Location
- a string of characters that, when interpreted via the Hypertext Transfer Protocol (HTTP)
- points to a data resource, notably files written in Hypertext Markup Language (HTML) or a subset of a database.

Some Basic Terminology: HTTP Methods / Verbs

- *GET*: requests a representation of a data resource corresponding to a particular URL. The process of executing the GET method is often referred to as a "GET request" and is the main method used for querying RESTful databases.
- *HEAD, POST, PUT, DELETE*: other common methods, though mostly never used for database querying.

How Do GET Requests Work? A Web Browsing Example

- Surfing the Web = Making a bunch of GET Requests
- For instance, I open my web browser and type in <http://www.wikipedia.org>. Once I hit return, I'd see a webpage.
- Several different processes occurred, however, between me hitting "return" and the page finally being rendered.

Step 1: The GET Request

- web browser took the entered character string
- used the command-line tool "Curl" to write a properly formatted HTTP GET request
- submitted it to the server that hosts the Wikipedia homepage.

STEP 2: The Response

- Wikipedia's server receives this request
- send back an HTTP response
- from which Curl extracted the HTML code for the page

```
[1] "<!DOCTYPE html>\n<html lang=\"mul\" dir=\"ltr\">\n<h
```

STEP 3: The Formatting

- raw HTML code was formatted and executed by the web browser
- rendering the page as seen in the window.

RESTful Database Querying: The GET Request

- URL we supply must be constructed so that the resulting request can be interpreted and successfully acted upon by the server.
- Likely that the character string must encode **search terms and/or filtering parameters**, as well as one or more **authentication codes**.
- While the terms are often similar across APIs, most are API-specific.

RESTful Database Querying: The Response

- unlike web browsing, the content of the server's response that is extracted by Curl is unlikely to be HTML code.
- will likely be **raw text** response that can be parsed into one of a few file formats commonly used for data storage.
- usual suspects include .csv, .xml, and .json files.

RESTful Database Querying: The Formatting

- web browser parsed the HTML code,
- but **we need R, Python, or other programming languages** to parse the server response
- and convert it into a format for local storage (e.g. matrices, dataframes, databases, lists, etc.).

Examples of Web APIs

- [Twitter APIs](#)
- [Data.gov APIs](#)
- [The Star Wars API \(SWAPI\)](#)
- [Google Dataset Search](#)

API Help(ers)!

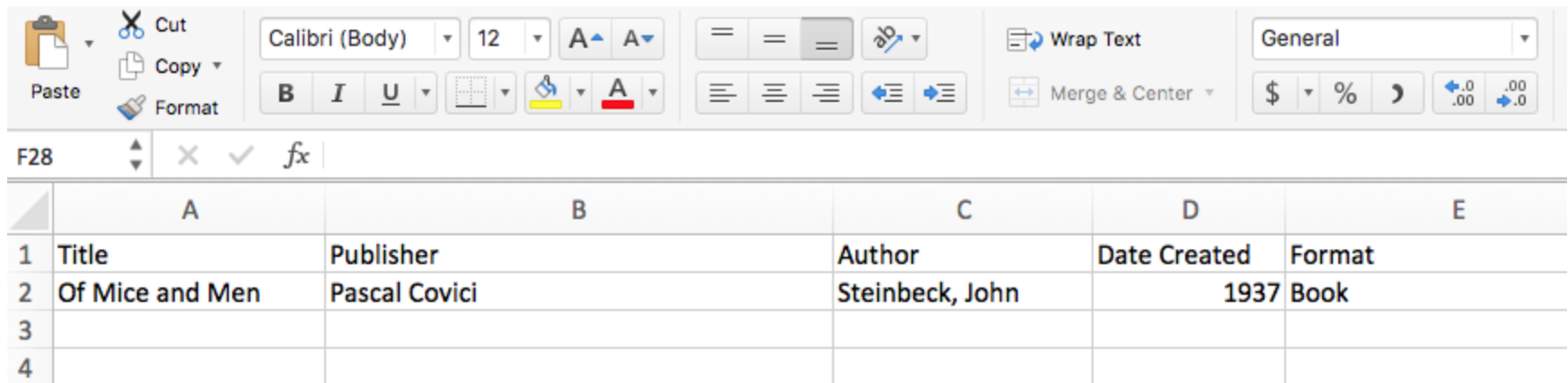
- [Twitter API Libraries](#)
 - [TwitterR](#)
- [Using Data.gov APIs in R](#)
- [SWAPI Helper libraries](#)

Intro to JSON

JSON Basics

- JavaScript Object Notation is a data format
- Based on a subset of the JavaScript Programming Language
- Text based and language independent

Simple JSON Example



The image shows the Microsoft Excel interface. The ribbon at the top includes the 'Home' tab with options for Paste, Cut, Copy, Format, font settings (Calibri, size 12), bold, italic, underline, text color, fill color, and alignment. The 'Wrap Text' and 'Merge & Center' buttons are also visible. Below the ribbon is the formula bar showing 'F28'. The spreadsheet contains a table with 5 columns (A-E) and 4 rows (1-4). The data in row 2 is: Title: 'Of Mice and Men', Publisher: 'Pascal Covici', Author: 'Steinbeck, John', Date Created: '1937', Format: 'Book'.

	A	B	C	D	E
1	Title	Publisher	Author	Date Created	Format
2	Of Mice and Men	Pascal Covici	Steinbeck, John	1937	Book
3					
4					

```
1 {  
2   "Title": "Of Mice and Men",  
3   "Publisher": "Pascal Covici",  
4   "Author": "Steinbeck, John",  
5   "DateCreated": "1937",  
6   "Format": "Book"  
7 }
```

JSON Data Types

- Strings: `"name" : "Jacob"`
- Numbers: `"age" : 30`
- Objects: `"employee" : {
 "name" : "Jacob" , "age" : 30 }`
- Arrays: `"employees" : ["Jacob", "Walt"]`
- Booleans: `"librarian" : true`

[More on JSON Data Types](#)

Complex JSON Example

```
{
  "paintings": [
    {
      "name": "The Scream",
      "url": "https://en.wikipedia.org/wiki/The_Scream",
      "creator": {
        "@type": "Person",
        "name": "Edvard Munch",
        "sameAs": "https://en.wikipedia.org/wiki/Edvard_Munch"
      }
    },
    {
      "name": "Melancholy",
      "url": "https://en.wikipedia.org/wiki/Melancholy_(Edvard_Munch)",
      "creator": {
        "@type": "Person",
        "name": "Edvard Munch",
        "sameAs": "https://en.wikipedia.org/wiki/Edvard_Munch"
      }
    }
  ]
}
```

JSON Help!

- [JSONLint](#)
- [\[Microsoft Excel\] Connect to a JSON File](#)
- Your JSON data is ready for analysis in Tableau 10.1!

Scavenger Hunt! Explore SWAPI

What is SWAPI?

swapi.co

- Star Wars API
- Structured data about episodic Star Wars films, up through The Force Awakens
- Info about films, people, planets, species, starships, vehicles (entities)

Navigating SWAPI

swapi.co

- Data organized into buckets
- <https://swapi.co/api/people> returns data about all people
- <https://swapi.co/api/people/1> returns data about the first person in the index, Luke Skywalker
- Each entity is connected to other entities by nesting URLs

```
{
  "name": "Luke Skywalker",
  "films": [
    "https://swapi.co/api/films/2/"
  ],
  "species": [
    "https://swapi.co/api/species/1/"
  ]
}
```

Instructions

Using your browser, explore SWAPI and answer these questions:

1. Which movies did George Lucas produce?
2. When was The Force Awakens released?
3. According to the API, which planets were featured in The Empire Strikes Back?
 - Which of these planets has the highest pop?

swapi.co

Next?

Questions?

Thank you!

Leigh Phan - leighphan@library.ucla.edu

Tim Dennis - timdennis@ucla.edu