

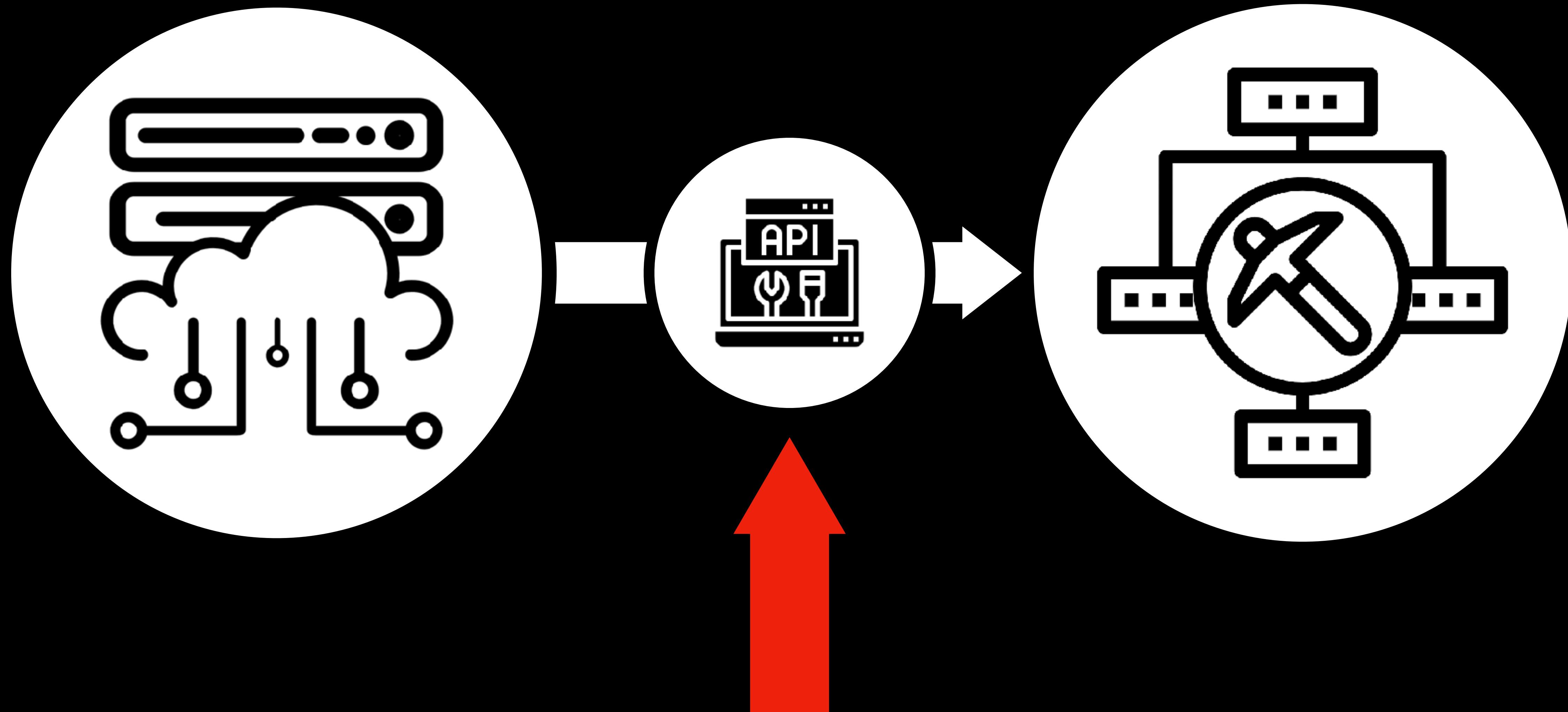
Intro to APIs

INST414 - Data Science Techniques

This Lecture's Learning Objectives

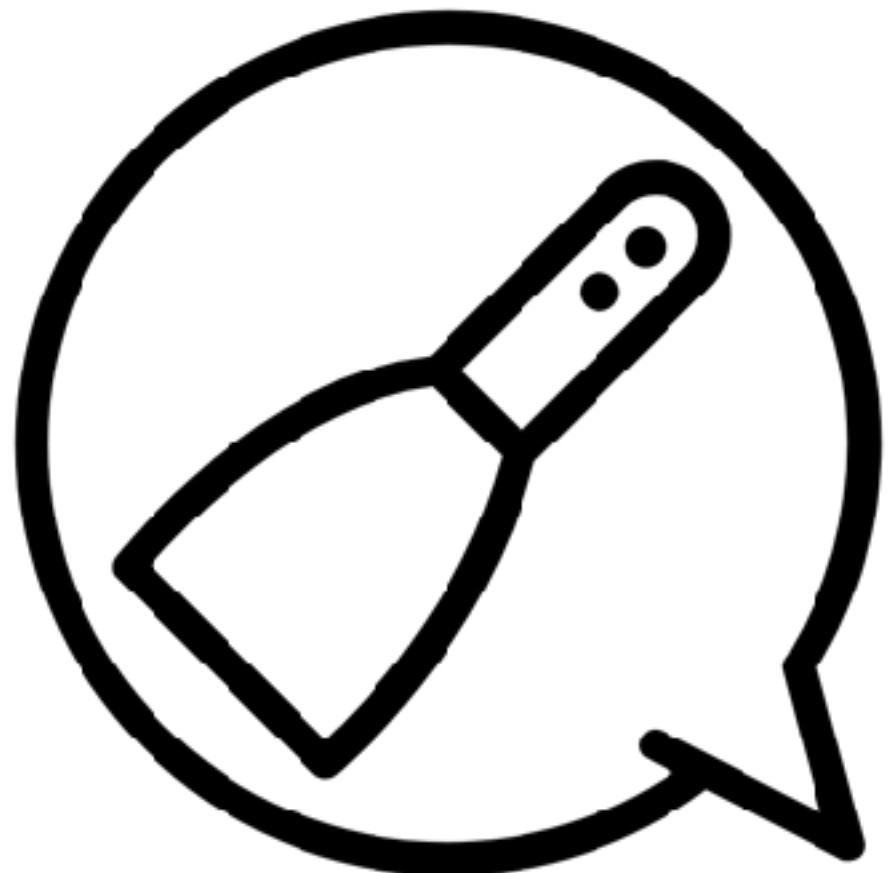
Describe at least two types of APIs and data interchange formats

This class: Mining + Analysis on Real Data

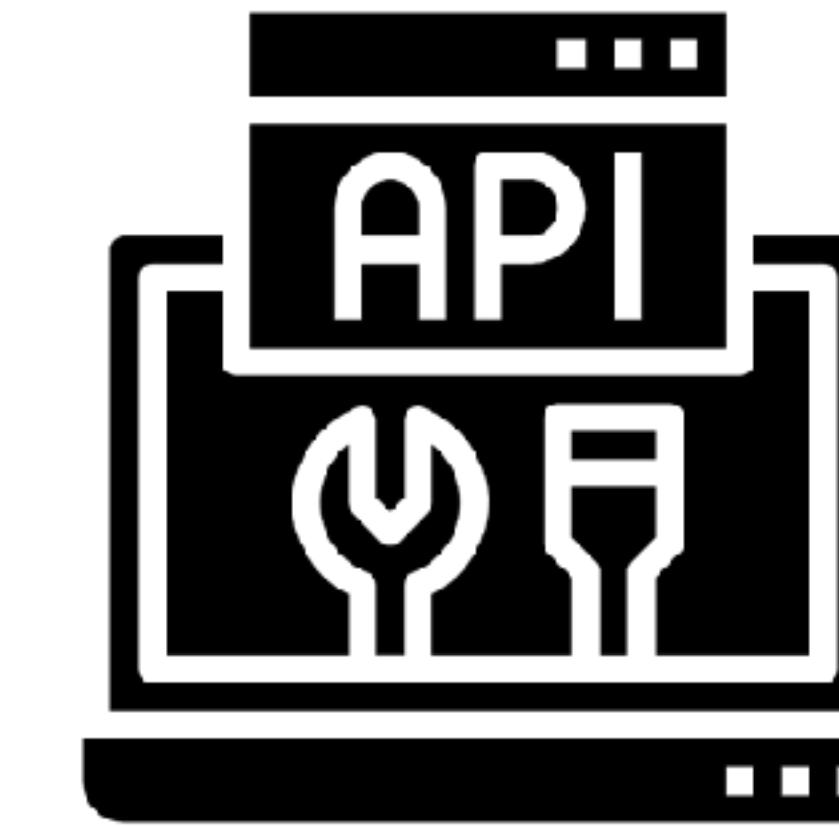


Motivating Question: How do we get data from the web?

Two Main Methods



Web Scraping



APIs

Two Main Methods



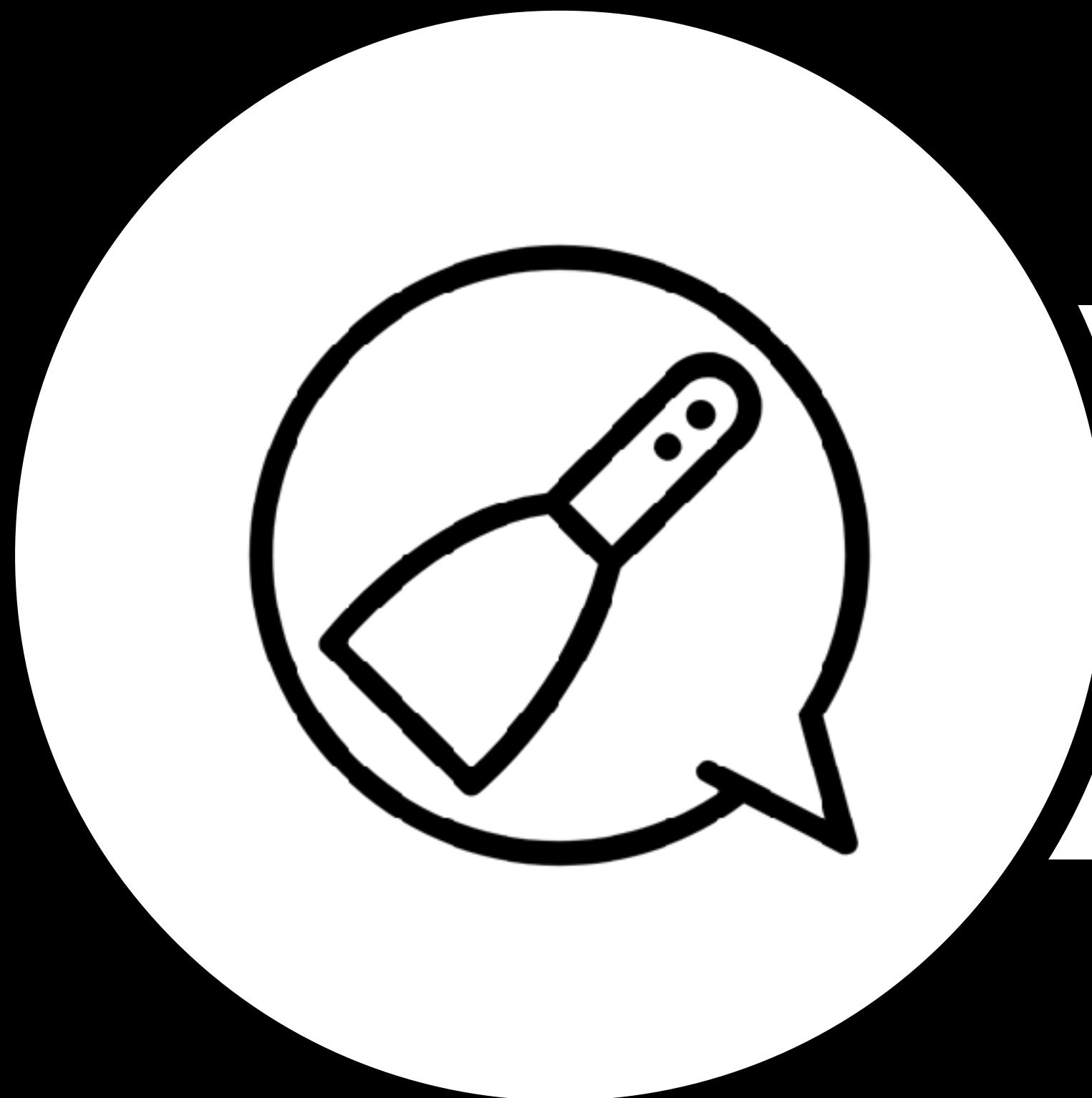
Web Scraping



APIs

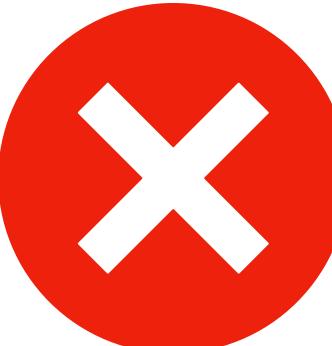


Advantages of Web Scraping

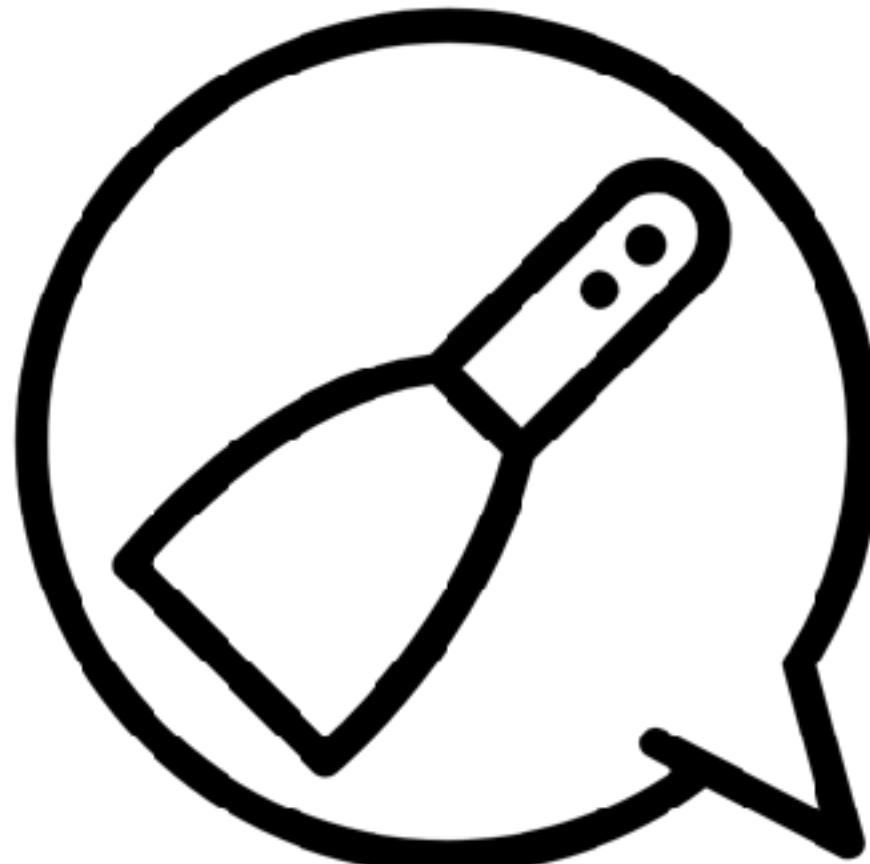


If you can view it on the web, you can scrape it

No API, so generally less limited by rate limits or auth



Disadvantages of Web Scraping



Very brittle; a slight UI change may break everything

Messy, unstructured data

May overload servers and get IP banned

Web Scraping Tools

The screenshot shows the Scrapy website (scrapy.org) in a browser. The page features a green header with the Scrapy logo and navigation links for Download, Documentation, Resources, Community, Commercial Support, FAQ, and Fork on GitHub. A prominent green box in the center says "Install the latest version of Scrapy" with a "Scrapy 2.4.1" badge and a "pip install scrapy" command. Below this are links for PayPal, Conda, and Release Notes. The main content area includes a brief description of Scrapy as an open-source framework for extracting data from websites, and a note that it is maintained by Scrapinghub and many other contributors. At the bottom, there's a link to the Beautiful Soup website.

An open source and collaborative framework for extracting the data you need from websites. In a fast, simple, yet extensible way.

Maintained by [Scrapinghub](#) and [many other contributors](#)

[Beautiful Soup: We called him "Tidy"](#)

[Download | Documentation | Hall of Fame | For enterprise | Source | Changelog | Discussion group | Zine]

Beautiful Soup

You didn't write that awful page. You're just trying to get some data out of it. Beautiful Soup is here to help. Since 2004, it's been saving programmers hours or days of work on quick-turnaround screen scraping projects.

Beautiful Soup is a Python library designed for quick turnaround projects like screen-scraping. Three features make it powerful:

1. Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn't take much code to write an application.
2. Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8. You don't have to think about encodings, unless the document doesn't specify an encoding and Beautiful Soup can't detect one. Then you just have to specify the original encoding.
3. Beautiful Soup sits on top of popular Python parsers like [lxml](#) and [html5lib](#), allowing you to try out different parsing strategies or trade speed for flexibility.

Beautiful Soup parses anything you give it, and does the tree traversal stuff for you. You can tell it "Find all the links", or "Find all the links of class `externalLink`", or "Find all the links whose urls match "`foo.com`", or "Find the table heading that's got bold text, then give me that text."

Valuable data that was once locked up in poorly-designed websites is now within your reach. Projects that would have taken hours take only minutes with Beautiful Soup.

Interested? [Read more.](#)

Getting and giving support

Beautiful Soup for enterprise available via Tidelift

If you have questions, send them to [the discussion group](#). If you find a bug, [file it on Launchpad](#). If it's a security vulnerability, report it confidentially through [Tidelift](#).

If you use Beautiful Soup as part of your work, please consider a [Tidelift subscription](#). This will support many of the free software projects your organization depends on, not just Beautiful Soup.

If Beautiful Soup is useful to you on a personal level, you might like to read [Tool Safety](#), a short zine I wrote about what I learned about software development from working on Beautiful Soup. Thanks!

Download Beautiful Soup

The current release is [Beautiful Soup 4.9.3](#) (October 3, 2020). You can install Beautiful Soup 4 with `pip install beautifulsoup4`.

The screenshot shows the SeleniumHQ website (selenium.dev) in a browser. The page has a black header with the Selenium logo and navigation links for About, Downloads, Projects, Documentation, Support, and Blog. A large black banner at the top reads "BLACK LIVES MATTER". Below it, text encourages users to support civil rights organizations. The main content area features a large green section with the heading "Selenium automates browsers. That's it!" and the subtext "What you do with that power is entirely up to you." Smaller text below explains that Selenium is for automating web applications for testing purposes and that boring web-based administration tasks can also be automated.

BLACK LIVES MATTER

In solidarity, we ask that you consider financially supporting efforts such as [The Equal Justice Initiative](#), [NAACP Legal Defense and Education Fund](#), or your local civil rights charity.

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.

Boring web-based administration tasks can (and should) also be automated as well.

Getting Started

Motivating Question: How do we get data from the web?

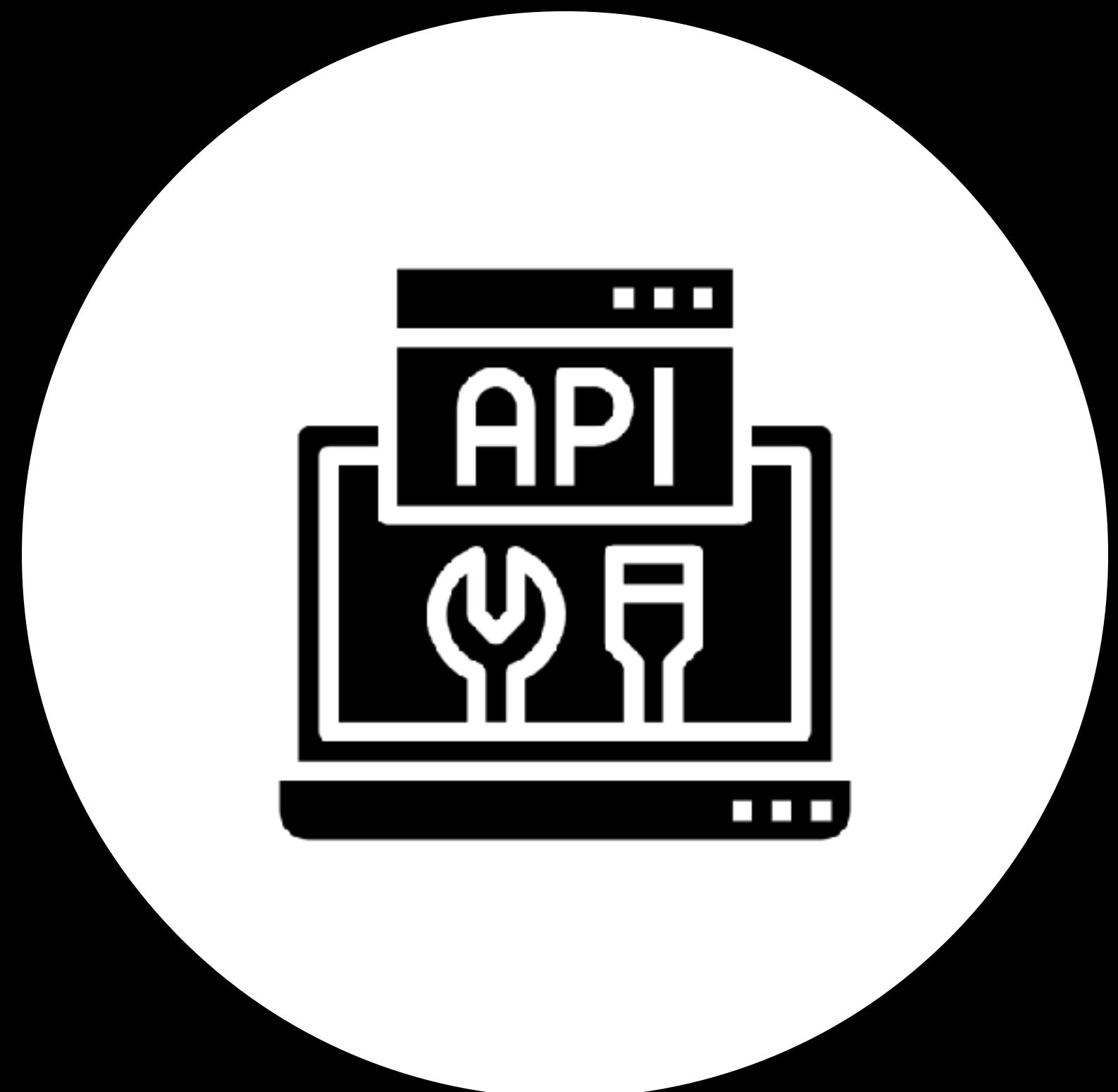
Use an API if available

Otherwise, can rely on scraping

What's an API?

“Application Programming Interface”

Different Types of Web APIs



Old-Style SOAP APIs

More Recent RESTful APIs

Any web-exposed interface meant for
programmatic access

Different Types of Web APIs

SOAP APIs

Request 1

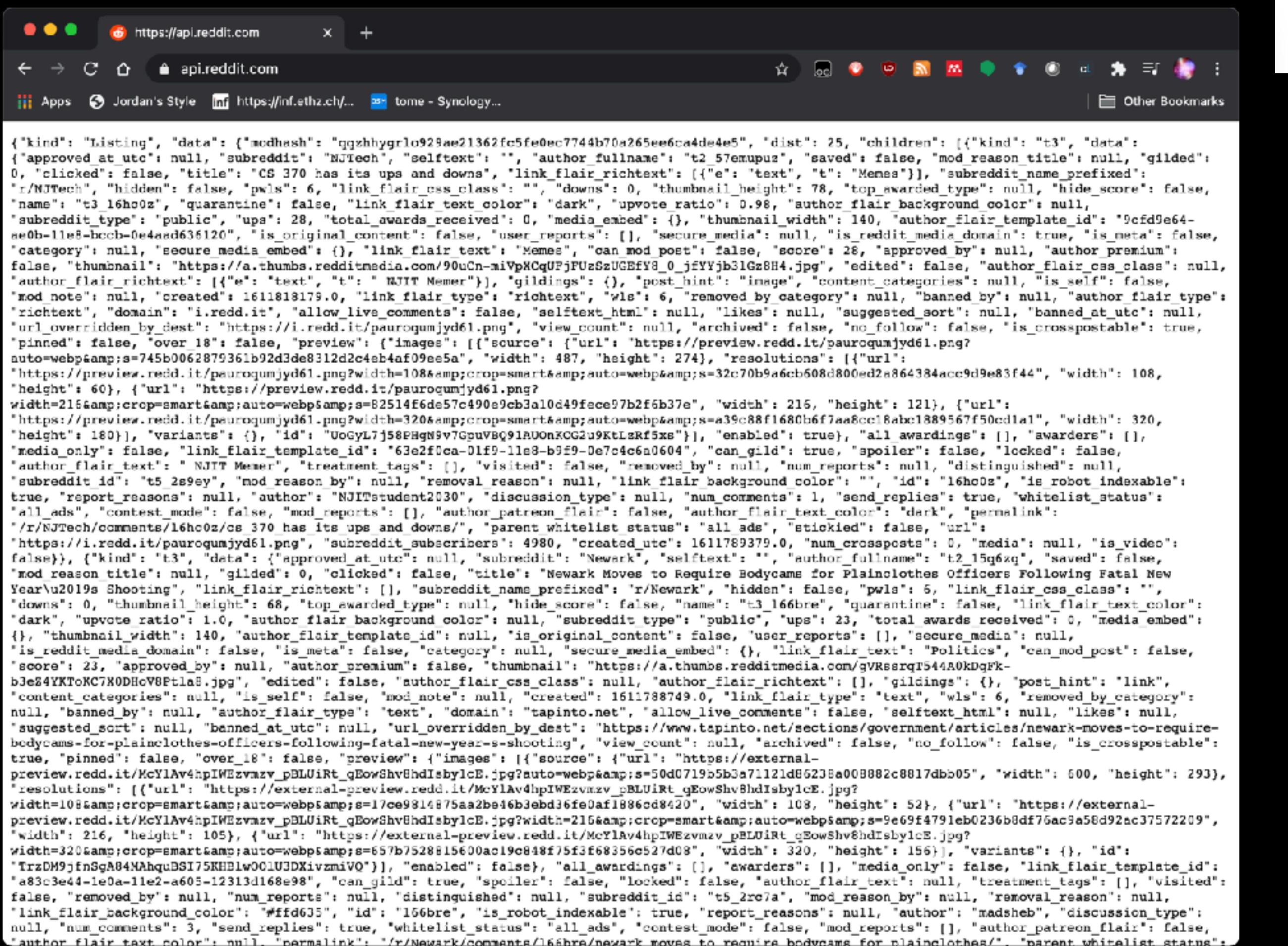
```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:enterprise.soap.sforce.com">
  <soapenv:Header>
    <urn:LoginScopeHeader>
      <urn:organizationId></urn:organizationId>
      <!--Optional:-->
      <urn:portalId></urn:portalId>
    </urn:LoginScopeHeader>
  </soapenv:Header>
  <soapenv:Body>
    <urn:login>
      <urn:username></urn:username>
      <urn:password></urn:password>
    </urn:login>
  </soapenv:Body>
</soapenv:Envelope>
```

Simple Objects Access Protocol

XML-based

Can (but need not) run over HTTP

Different Types of Web APIs



A screenshot of a web browser window showing a JSON API response from <https://api.reddit.com>. The response is a complex object containing various fields such as kind, data, children, title, author, flair, and many others. The JSON is heavily nested and includes URLs for thumbnails and other API endpoints.

```
{"kind": "Listing", "data": {"modhash": "qgzhhggrlo929ae21362fc5fe0ec7744b70a265ee6ca4de4e5", "dist": 25, "children": [{"kind": "t3", "data": {"approved_at_utc": null, " subreddit": "NJTech", "selftext": "", "author_fullname": "t2_57emupuz", "saved": false, "mod_reason_title": null, "gilded": 0, "clicked": false, "title": "CS 370 has its ups and downs", "link_flair_richtext": [{"e": "text", "t": "Memes"}], " subreddit_name_prefixed": "r/NJTech", "hidden": false, "pws": 6, "link_flair_css_class": "", "downs": 0, " thumbnail_height": 78, "top_awarded_type": null, "hide_score": false, "name": "t3_16hc0z", "quarantine": false, "link_flair_text_color": "dark", "upvote_ratio": 0.98, "author_flair_background_color": null, " subreddit_type": "public", "ups": 28, "total_awards_received": 0, "media_embed": {}, " thumbnail_width": 140, "author_flair_template_id": "9cf9d9e64-e0b-11e8-bccb-0e4ad636120", "is_original_content": false, "user_reports": {}, "secure_media": null, "is_reddit_media_domain": true, "is_meta": false, "category": null, "secure_media_embed": {}, "link_flair_text": "Memes", "can_mod_post": false, "score": 28, "approved_by": null, "author_premium": false, " thumbnail": "https://a.thumbs.redditmedia.com/90uCn-miVpXQgUPjPUz6zUGBFY8_0_jfYYjb31Gz8H4.jpg", "edited": false, "author_flair_css_class": null, "author_flair_richtext": [{"e": "text", "t": " NJIT Memer"}], "gildings": {}, "post_hint": "image", "content_categories": null, "is_self": false, "mod_note": null, "created": 1611818179.0, "link_flair_type": "richtext", "wls": 6, "removed_by_category": null, "banned_by": null, "author_flair_type": "richtext", "domain": "i.redd.it", "allow_live_comments": false, "selftext_html": null, "likes": null, "suggested_sort": null, "banned_at_utc": null, "url_overridden_by_dest": "https://i.redd.it/pauroqumjyd61.png", "view_count": null, "archived": false, "no_follow": false, "is_crosspostable": true, "pinned": false, "over_18": false, "preview": {"images": [{"source": {"url": "https://preview.reddit.it/pauroqumjyd61.png?width=487, height=274}, "resolutions": [{"url": "https://preview.reddit.it/pauroqumjyd61.png?width=108&crop=smart&auto=webp&s=32c70b9af6cb608d800ed2a864384acc9d9e83f44", "width": 108, "height": 60}, {"url": "https://preview.reddit.it/pauroqumjyd61.png?width=215&crop=smart&auto=webp&s=82514f6dc57c490a9cb3a10d49face97b2f6b37e", "width": 215, "height": 121}, {"url": "https://preview.reddit.it/pauroqumjyd61.png?width=320&crop=smart&auto=webp&s=a39c88f1680b6f7aa8cc18abc1889567f50cd1a1", "width": 320, "height": 180}], "variants": {}, "id": "UoGyL7j58FBgN9v7GsuVBg91AUOnKCG2u9ktLsRf5xe"}, "enabled": true, "all_awardings": [], "awarders": [], " media_only": false, "link_flair_template_id": "63e2f1ca-01f9-11e8-b3f9-0e7c4c6a0604", "can_gild": true, "spoiler": false, "locked": false, "author_flair_text": " NJIT Memer", "treatment_tags": [], "visited": false, "removed_by": null, "num_reports": null, "distinguished": null, " subreddit_id": "t5_2s9ey", "mod_reason_by": null, "removal_reason": null, "link_flair_background_color": "", "id": "16hc0z", "is_robot_indexable": true, "report_reasons": null, "author": "NJITstudent2030", "discussion_type": null, "num_comments": 1, "send_replies": true, "whitelist_status": "all_ads", "contest_mode": false, "mod_reports": [], "author_patreon_flair": false, "author_flair_text_color": "dark", "permalink": "/r/NJTech/comments/16hc0z/cs_370 has its ups and downs/", "parent_whitelist_status": "all_ads", "stickied": false, "url": "https://i.redd.it/pauroqumjyd61.png", " subreddit_subscribers": 4980, "created_utc": 1611789379.0, "num_crossposts": 0, "media": null, "is_video": false}, {"kind": "t3", "data": {"approved_at_utc": null, " subreddit": "Newark", "selftext": "", "author_fullname": "t2_15g6xq", "saved": false, "mod_reason_title": null, "gilded": 0, "clicked": false, "title": "Newark Moves to Require Bodycams for Plainclothes Officers Following Fatal New Year\u2019s Shooting", "link_flair_richtext": [], " subreddit_name_prefixed": "r/Newark", "hidden": false, "pws": 5, "link_flair_css_class": "", "downs": 0, " thumbnail_height": 68, "top_awarded_type": null, "hide_score": false, "name": "t3_166bre", "quarantine": false, "link_flair_text_color": "dark", "upvote_ratio": 1.0, "author_flair_background_color": null, " subreddit_type": "public", "ups": 23, "total_awards_received": 0, "media_embed": {}, " thumbnail_width": 140, "author_flair_template_id": null, "is_original_content": false, "user_reports": {}, "secure_media": null, "is_reddit_media_domain": false, "is_meta": false, "category": null, "secure_media_embed": {}, "link_flair_text": "Politics", "can_mod_post": false, "score": 23, "approved_by": null, "author_premium": false, " thumbnail": "https://a.thumbs.redditmedia.com/qVRssrqT544A0KDqFk-ble24YKToKC7XODHcV8Ptlas.jpg", "edited": false, "author_flair_css_class": null, "link_flair_richtext": [], "gildings": {}, "post_hint": "link", "content_categories": null, "is_self": false, "mod_note": null, "created": 1611788749.0, "link_flair_type": "text", "wls": 6, "removed_by_category": null, "banned_by": null, "author_flair_type": "text", "domain": "tapinto.net", "allow_live_comments": false, "selftext_html": null, "likes": null, "suggested_sort": null, "banned_at_utc": null, "url_overridden_by_dest": "https://www.tapinto.net/sections/government/articles/newark-moves-to-require-bodycams-for-plainclothes-officers-following-fatal-new-year-s-shooting", "view_count": null, "archived": false, "no_follow": false, "is_crosspostable": true, "pinned": false, "over_18": false, "preview": {"images": [{"source": {"url": "https://external-preview.reddit.it/McYlAv4hpIWEzvmzv_pBLUiRt_qBowShv8hdIsby1cE.jpg?auto=webp&s=50d0719b5b3a71121d86235a008882c8817dbb05", "width": 600, "height": 293}, "resolutions": [{"url": "https://external-preview.reddit.it/McYlAv4hpIWEzvmzv_pBLUiRt_qBowShv8hdIsby1cE.jpg?width=108&crop=smart&auto=webp&s=1ce9314875aa2be16b3ebd36fe0af1886cd8420", "width": 108, "height": 53}, {"url": "https://external-preview.reddit.it/McYlAv4hpIWEzvmzv_pBLUiRt_qBowShv8hdIsby1cE.jpg?width=216&crop=smart&auto=webp&s=969f4791eb0236b8df75ac9a58d92ac37572209", "width": 216, "height": 105}, {"url": "https://external-preview.reddit.it/McYlAv4hpIWEzvmzv_pBLUiRt_qBowShv8hdIsby1cE.jpg?width=320&crop=smart&auto=webp&s=657b752815600ac19c88f75f3f68356c52d087", "width": 320, "height": 156}], "variants": {}, "id": "TrzDM9jfnSgA84NAhquBSI75KHB1w0C1UDXKivzniVO}], "enabled": false, "all_awardings": [], "awarders": [], " media_only": false, "link_flair_template_id": "a83c3e44-1e0a-11e2-a605-12313d168e98", "can_gild": true, "spoiler": false, "locked": false, "author_flair_text": null, "treatment_tags": [], "visited": false, "removed_by": null, "num_reports": null, "distinguished": null, " subreddit_id": "t5_2rc7a", "mod_reason_by": null, "removal_reason": null, "link_flair_background_color": "#ffd635", "id": "166bre", "is_robot_indexable": true, "report_reasons": null, "author": "madsheb", "discussion_type": null, "num_comments": 3, "send_replies": true, "whitelist_status": "all_ads", "contest_mode": false, "mod_reports": [], "author_patreon_flair": false, "author_flair_text_color": null, "permalink": "/r/Newark/comments/166bre/newark_moves_to_require_bodycams_for_plainclothes/"}, "parent_whitelist_status": "
```

RESTful APIs

REpresentational State Transfer

JSON-based

Meant to operate on top of HTTP

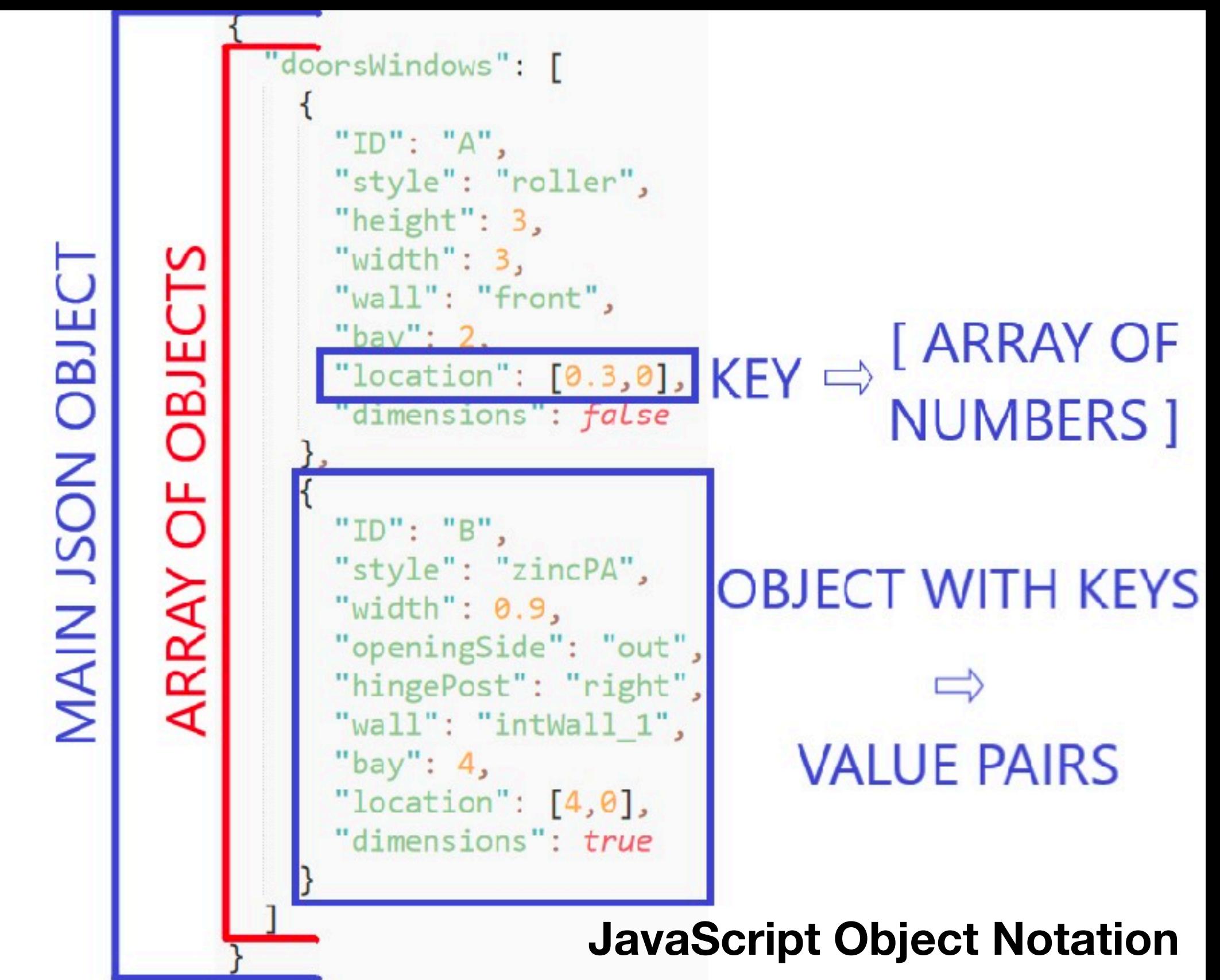
Avoids complex firewalls

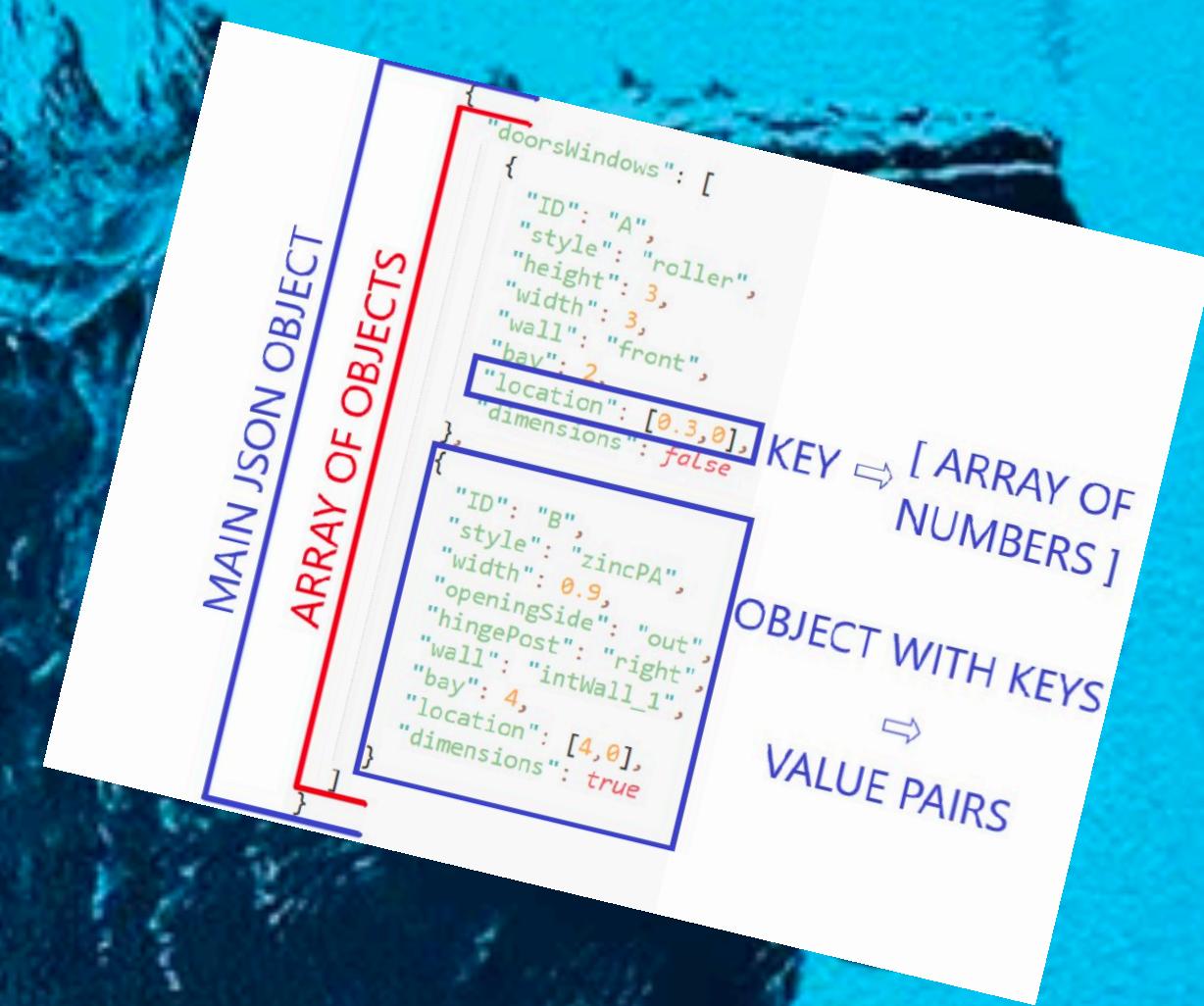
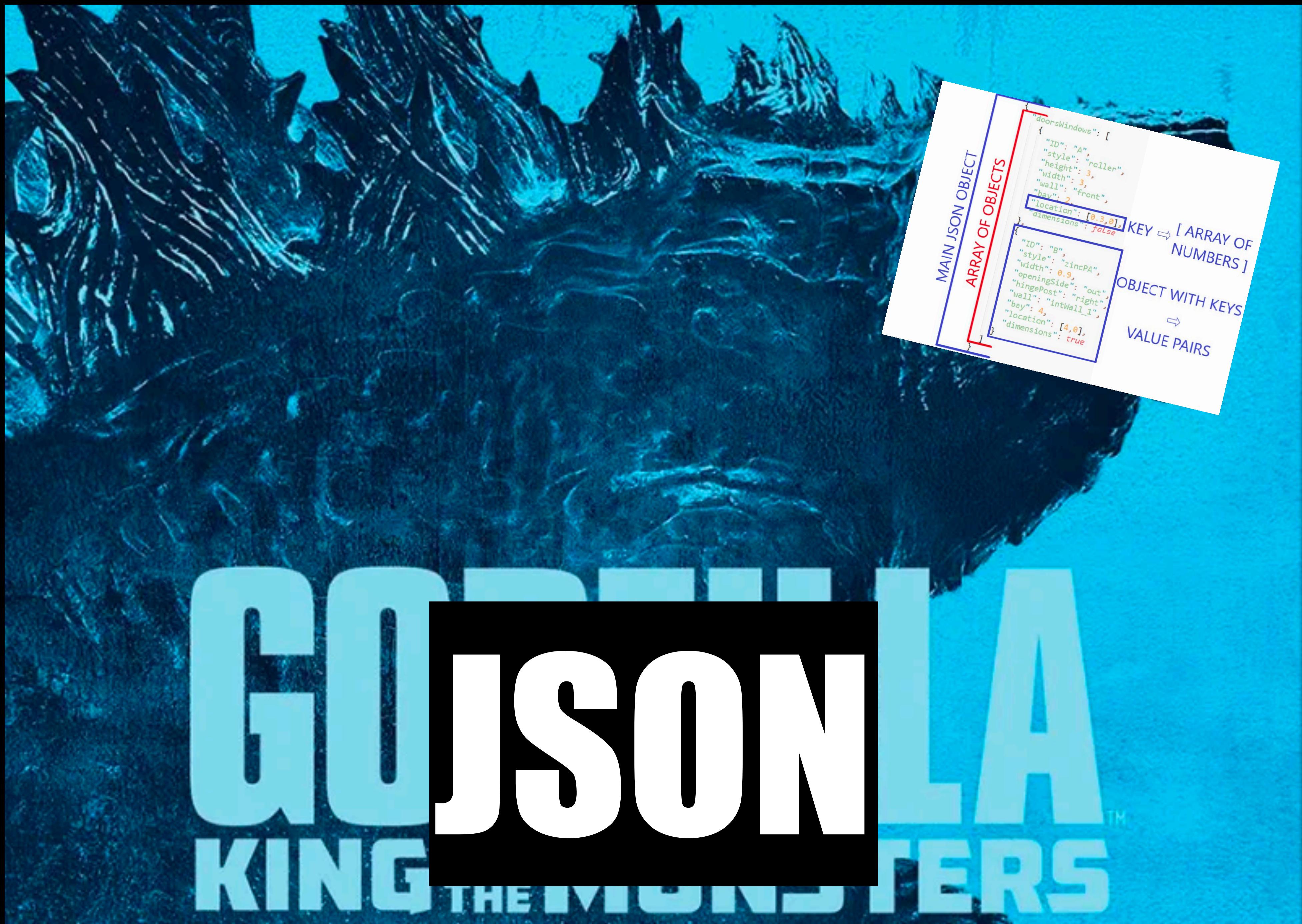
XML vs. JSON

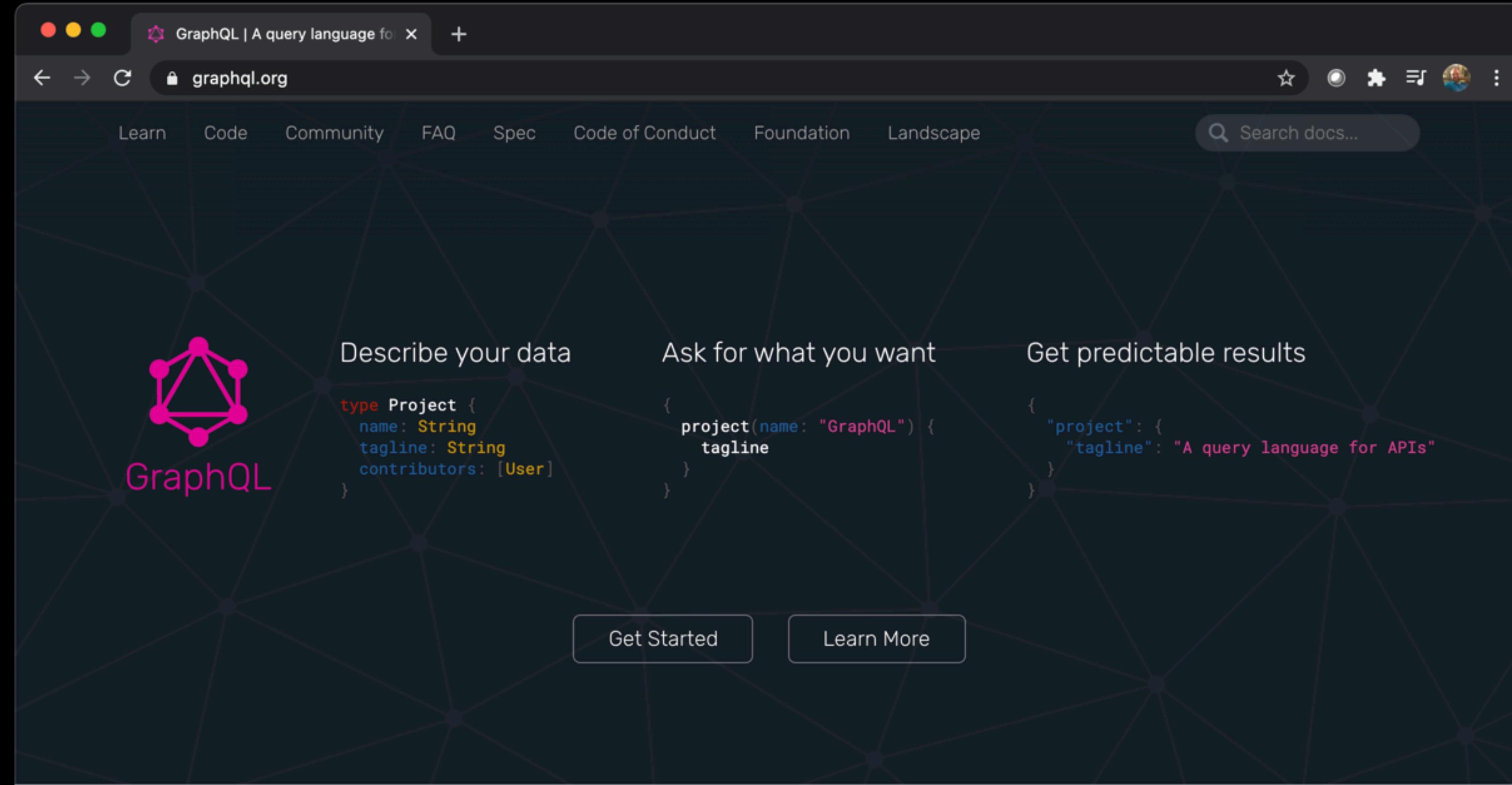
Both types of data interchange formats

```
- <Parts>
  - <Part>
    <Id>4478</Id>
    <Part_Name>1000 Ohm Resistor</Part_Name>
    <Total_Available>25000</Total_Available>
    <Price>0.01</Price>
  </Part>
  - <Part>
    <Id>3328</Id>
    <Part_Name>15000 Ohm Resistor</Part_Name>
    <Total_Available>75000</Total_Available>
    <Price>0.02</Price>
  </Part>
  - <Part>
    <Id>4725</Id>
    <Part_Name>555 Timer IC</Part_Name>
    <Total_Available>1500</Total_Available>
    <Price>0.25</Price>
  </Part>
</Parts>
```

eXtensible Markup Language







A query language for your API

GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and

unambiguous description of the data in your API, so clients can efficiently request exactly what they need.

Some quick background on HTTP

HyperText Transfer Protocol

INST414-0103: Data Science Techniques - Spring 2023

https://umd.instructure.com/courses/1340623

UNIVERSITY OF MARYLAND

Account

Dashboard

Courses

Calendar

Inbox

Portfolium

History

CourseExp

INST414 > Modules

Spring 2023

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Quizzes

Modules

BigBlueButton

Collaborations

Chat

Course Logistics

Participation - Introductions

Jan 27 | 100 pts

Class Preparation

Pre-Class Exercise - Most Frequent Tokens in Text

Jan 30 | 100 pts

Resources

Syllabus

Previous Semester's YouTube Playlist

Collapse All

24/7 Canvas Chat Support

...or call 1-833-566-3347 (staff/faculty)
1-877-399-4090 (students)

View Course Stream

View Course Calendar

View Course Notifications

To Do

Project Ideas

INST414-0103: Data Science Techniques - Spring 2023 cbuntain Jan 23 at 11:59pm

Week 1 Slides Posted

INST414-0103: Data Science Techniques - Spring 2023 cbuntain Jan 25 at 1:33pm

Participation - Introductions

HyperText Transfer Protocol, Initially RFC1945

Screenshot of the IETF DataTracker website showing the RFC 1945 document page.

Network Working Group
Request for Comments: 1945
Category: Informational

T. Berners-Lee
MIT/LCS
R. Fielding
UC Irvine
H. Frystyk
MIT/LCS
May 1996

Hypertext Transfer Protocol -- HTTP/1.0

Status of This Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

IESG Note:

The IESG has concerns about this protocol, and expects this document to be replaced relatively soon by a standards track document.

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification reflects common usage of the protocol referred to as "HTTP/1.0".

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 Terminology	4
1.3 Overall Operation	6
1.4 HTTP and MIME	8
2. Notational Conventions and Generic Grammar	8
2.1 Augmented BNF	8
2.2 Basic Rules	10
3. Protocol Parameters	12

Hypertext Transfer Protocol



International standard [RFC 1945](#) HTTP/1.0 (1996)

[RFC 2616](#) HTTP/1.1 (1999)

[RFC 7540](#) HTTP/2 (2015)

[RFC 7541](#) Header Compression (2, 2015)

[RFC 7230](#) Message Syntax and Routing (1.1, 2014)

[RFC 7231](#) Semantics and Content (1.1, 2014)

[RFC 7232](#) Conditional Requests (1.1, 2014)

[RFC 7233](#) Range Requests (1.1, 2014)

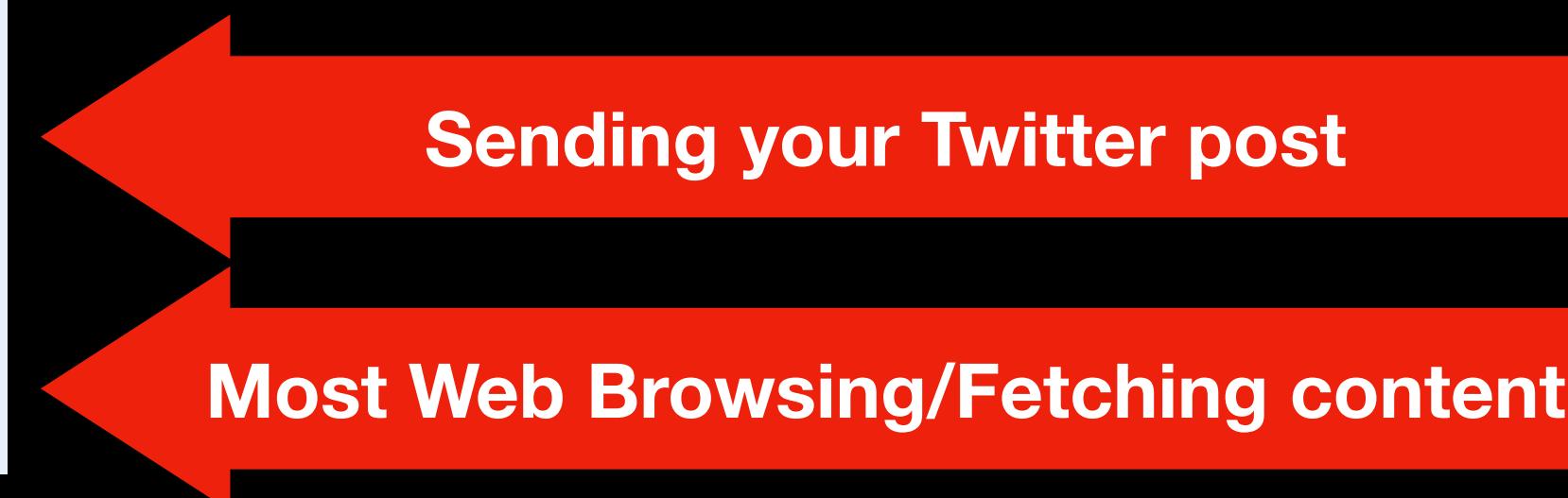
[RFC 7234](#) Caching (1.1, 2014)

[RFC 7235](#) Authentication (1.1, 2014)

Developed by initially [CERN](#); [IETF](#), [W3C](#)

Introduced 1991; 30 years ago

HTTP method	Description
Head	To send header part data
Trace	To resend the receive request
Put	Help us to put file into server
Delete	Remove file from server
Options	It determines which of the HTTP method server support and return appropriate error
Post	It is used to post data into server
Get	It used to get data from server



The screenshot shows a Mac OS X desktop with a dark-themed browser window. The title bar reads "Requests: HTTP for Humans™" and the URL is "requests.readthedocs.io/en/master/". The page content includes a "Star" button with "44,453" stars, a search bar, and a brief description of the Requests library. It also features a "Fork me on GitHub" badge and a "Sponsored by CERT Gouvernemental - GOVCERT.LU" section. A red box highlights a code snippet demonstrating the library's usage:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type": "User"...'
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

Below the code, there's a link to "similar code, sans Requests." A paragraph explains the library's ease of use and its reliance on `urllib3`. The "Useful Links" sidebar on the left lists various documentation sections and external links.

Beloved Features

Requests is ready for today's web.

- Keep-Alive & Connection Pooling
- International Domains and URLs
- Sessions with Cookie Persistence
- Browser-style SSL Verification

Python's HTTP-based Requests library has good support for working with APIs

Get started using the APIs in four steps:

Create an Account

If you don't already have an account create one:

1. Click [Sign In](#).
2. Click [Create account](#) and follow prompts to register.

Sign In

To sign in to the portal:

1. Click [Sign In](#).
2. Enter your email address and password.
3. Click [Sign In](#).

Register apps

To register an app:

1. Select **My Apps** from the user drop-down.
2. Click **+ New App** to create a new app.
3. Enter a name and description for the app in the New App dialog.
4. Click **Create**.
5. Click the **APIs** tab.
6. Click the access toggle to enable or disable access to an API product from the app.

APIs allow services to enforce access control and rate-limits

Access the API keys

1. Select **My Apps** from the user drop-down.
2. Click the app in the list.
3. View the API key on the **App Details** tab.
4. Confirm that the status of the API key is Approved.

APIs

Often have to register to get developer keys

Restricted to no more than X requests per minute

Fortunately, don't often need to rely on raw HTTP for API access

Many popular services provide API wrappers

Twitter API Documentation | Doc X +

developer.twitter.com/en/docs/twitter-api

Developer Documentation

Search the docs

Twitter API > Twitter API

- Getting started
- Tutorials
- Tools and libraries
- Migrate
- API reference index

The new Twitter API v2

- Fundamentals
- Tweets
- Users

Enterprise Gnip 2.0

Twitter API

Programmatically analyze and engage with the conversation happening on Twitter.

The Twitter API provides the tools you need to contribute to and engage with the conversation happening on Twitter.

Get started >

Twitter API v2: Early Access

We're building a new Twitter API with a modern and more improved developer experience. Early Access is now available and control the conversation on Twitter. Additional endpoints will be released soon!

Twitter API Tools & Libraries | Doc X +

developer.twitter.com/en/docs/twitter-api/tools-and-libraries

Developer Documentation

Search the docs

Twitter API > Twitter API

- Getting started
- Tutorials
- Tools and libraries
- Migrate
- API reference index

The new Twitter API v2

- Fundamentals
- Tweets
- Users

Enterprise Gnip 2.0

Official tools and libraries

The Twitter teams maintain a set of official libraries and SDKs, listed here. We also include a list of [community-supported libraries](#) lower on this page.

JavaScript / Node.js

Clients SDKs / Libraries Tools

[Autohook](#)
Get started with the [Premium v1.1](#) Account Activity API

Python

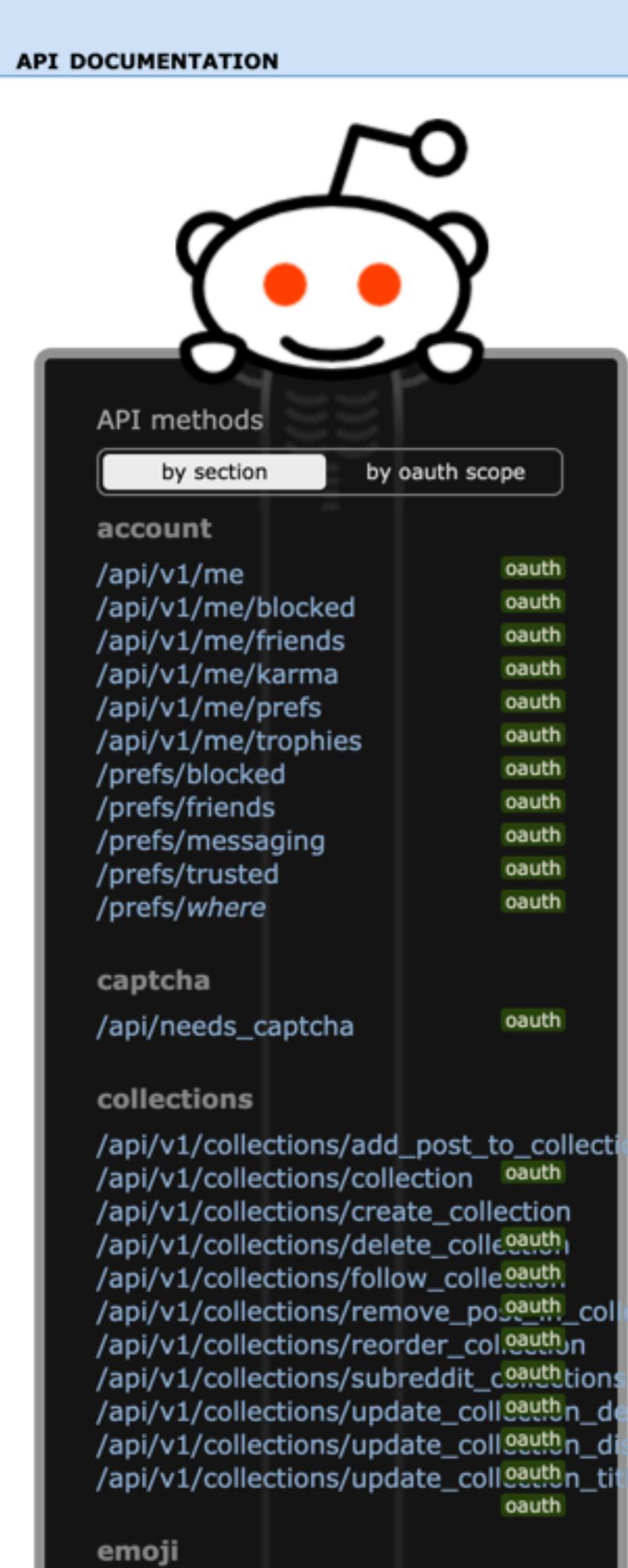
Clients SDKs / Libraries Tools

[search-tweets-python](#)
A client supporting [v2](#), [Premium v1.1](#) and [Enterprise](#) search

Ruby

Clients SDKs / Libraries Tools

[search-tweets-ruby](#)



This is automatically-generated documentation for the reddit API.

Please take care to respect our [API access rules](#).

overview

listings

Many endpoints on reddit use the same protocol to return data. These endpoints are called Listings and share five common parameters: `before`, `limit`, `count`, and `show`.

Listings do not use page numbers because their data is ordered by relevance. They allow you to view slices of the underlying data using the `after` and `before` fields which are equivalent to `since` and `before` on the site and in combination with `count` can be used to page through a listing.

The common parameters are as follows:

- `after` / `before` - only one should be provided. It specifies an item in the listing to use as the anchor point for the next request.
- `limit` - the maximum number of items to return.
- `count` - the number of items already seen. The PRAW builder uses this to determine when to give up trying to fetch more items.
- `show` - optional parameter; if `all` is provided, "not voted on" and "not have voted on" will be disabled.

To page through a listing, start by fetching the first few items using `after` and `count`. The response will contain `after` and `count` in the next request. It is a good idea, but not required, to provide the same `count` which should be the number of items already seen.

modhashes

A modhash is a token that the reddit API requires to be included with every request. It can be obtained via the [/api/me.json](#) call or in responses to other API calls.

The preferred way to send a modhash is to include it with your requests.

Modhashes are not required when authenticated with OAuth 2.0.

PRAW: The Python Reddit API

PRAW's documentation

- [Getting Started](#)
- [Code Overview](#)
- [Tutorials](#)
- [Package Info](#)

Documentation

Unless otherwise mentioned, see [Authenticating via OAuth 2.0](#).

Getting Started

- [Quick Start](#)
- [Installing PRAW](#)
- [Authenticating via OAuth 2.0](#)
- [Configuring PRAW](#)
- [Running Multiple Instances of PRAW](#)
- [Logging in PRAW](#)
- [Frequently Asked Questions](#)

Code Overview

- [The Reddit Instance](#)
- [Working with PRAW's Models](#)
- [Exceptions in PRAW](#)
- [Other Classes](#)

Tutorials

- [Comment Extraction and Parsing](#)
- [Obtaining a Refresh Token](#)
- [Submission Stream Reply Bot](#)

Package Info

- [Change Log](#)
- [Contributing to PRAW](#)
- [Glossary](#)

PRAW: The Python Reddit API

praw.readthedocs.io/en/latest/

Docs » PRAW: The Python Reddit API Wrapper

Edit on GitHub

PRAW: The Python Reddit API Wrapper

PRAW's documentation is organized into the following sections:

- [Getting Started](#)
- [Code Overview](#)
- [Tutorials](#)
- [Package Info](#)

Documentation Conventions

Unless otherwise mentioned, all examples in this document assume the use of a `script` application. See [Authenticating via OAuth](#) for information on using `installed` applications and `web` applications.

Getting Started

- [Quick Start](#)
- [Installing PRAW](#)
- [Authenticating via OAuth](#)
- [Configuring PRAW](#)
- [Running Multiple Instances of PRAW](#)
- [Logging in PRAW](#)
- [Frequently Asked Questions](#)

Code Overview

- [The Reddit Instance](#)
- [Working with PRAW's Models](#)
- [Exceptions in PRAW](#)
- [Other Classes](#)

Tutorials

- [Comment Extraction and Parsing](#)
- [Obtaining a Refresh Token](#)
- [Submission Stream Reply Bot](#)

Package Info

- [Change Log](#)
- [Contributing to PRAW](#)
- [Glossary](#)

praw api - Google Search X APIs and SDKs - Graph API X +

← → C 🔒 developers.facebook.com/docs/apis-and-sdks

FACEBOOK for Developers Docs Tools Support Search developer documentation Log In

Graph API

- Overview
- Using the Graph API
- Graph Explorer Guide
- Versioning
- Upgrade Guide
- Common Uses
- Handling Errors
- Debugging
- APIs and SDKs**
- Using Our SDKs
- FAQ
- Reference
- Features Reference
- Permissions Reference
- Webhooks
- Upload Files to Facebook with Resumable Upload API
- Advanced
- Changelog
- Server-Sent Events

On This Page

APIs and SDKs

Find out how our APIs work and integrate our products quickly using SDKs for common languages and platforms.



Our APIs

Graph API
The primary way for apps to read and write to the Facebook social graph. Access Pages, Users, Posts, Groups, Events and more.

Marketing APIs
The Marketing APIs offer programmatic access to your Facebook ad campaigns, Creative Materials, and reports.

Our Official SDKs

Facebook SDK for Android
Track installs and app opens, give people the ability to share content, or support the ability to Login with Facebook.

Facebook Business SDK
If you're using the Ads API, this makes it easy to manage your Facebook business account, Custom Audiences, and reports.

Facebook SDK for iOS
The easiest way to track the success of your App Ads, measure user engagement, and build social features.

Facebook SDK for React
Helps you build engaging social experiences with React.

Guides

Examples
Try our examples for using the SDK: Trigger a Share dialog, Facebook Login and calling the Graph API.

Advanced Setup
Learn about customizing options of the Facebook SDK for JavaScript.

Frameworks

AngularJS
Concepts how to integrate the Facebook SDK for AngularJS.

jQuery
Incorporate the Facebook SDK for JavaScript into your jQuery applications.

JavaScript SDK - Web SDKs X +

← → C 🔒 developers.facebook.com/docs/javascript/

FACEBOOK for Developers Docs Tools Support Search developer documentation Log In

Facebook SDK for JavaScript



Web SDKs

- JavaScript SDK
- Quickstart
- Advanced Setup
- Examples
- Frameworks
- Reference

On This Page

Quickstart
Learn how to use the JavaScript SDK.

Reference
A list of all SDK methods.

Guides

Examples
Try our examples for using the SDK: Trigger a Share dialog, Facebook Login and calling the Graph API.

Advanced Setup
Learn about customizing options of the Facebook SDK for JavaScript.

Frameworks

AngularJS
Concepts how to integrate the Facebook SDK for AngularJS.

jQuery
Incorporate the Facebook SDK for JavaScript into your jQuery applications.

The image shows two browser windows side-by-side. The left window displays the [New York Times Developers](https://developer.nytimes.com) website. The right window displays the [pynytimes](https://pypi.org/project/pynytimes/) project page on PyPI.

New York Times Developers Website (Left):

- Header:** praw api - Google Search, Home | Dev Portal, developer.nytimes.com
- Title:** {T} Developers
- Main Section:** The New York Times Developer Network, All the APIs Fit to Post
- Buttons:** GET STARTED
- Sections:** Get Started, APIs, FAQ, Branding, Covid-19 Data

pynytimes Project Page on PyPI (Right):

- Header:** pynytimes · PyPI, pypi.org/project/pynytimes/
- Title:** pynytimes 0.4.2
- Install Link:** pip install pynytimes
- Release Date:** Released: Dec 28, 2020
- Description:** A Python wrapper for (most) New York Times APIs
- Navigation:** Project description, Release history, Download files
- Project links:** Homepage
- Statistics:** GitHub statistics, Stars: 6
- Installation:** There are multiple options to install and upgrade pynytimes, but the easiest is by just installing it using pip (or pip3).
- Linux and Mac:** pip install --upgrade pynytimes

GitHub - public-apis/public-api

github.com/public-apis/public-apis

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search Sign in Sign up

public-apis / public-apis

Code Issues 4 Pull requests 236 Actions Projects Security Insights

master 2 branches 0 tags Go to file Code

yannbertrand Remove Barchart 5a39032 18 days ago 2,270 commits

.github Add some information on CORS (#1446) 3 months ago

build Faking user agent + handle "no server" and fallback error 2 months ago

.gitattributes Ignore .github 4 years ago

.travis.yml Update build scripts to pass flake8 3 years ago

README.md Remove Barchart 18 days ago

README.md

Public APIs

Run tests passing Validate links failing

A collective list of free APIs for use in software and web development.

A public API for this project can be found [here!](#)

For information on contributing to this project, please see the [contributing guide](#).

Please note a passing build status indicates all listed APIs are available since the last update. A failing build status indicates that 1 or more services may be unavailable at the moment.

Watch 3.1k Star 109k Fork 13.4k

About

A collective list of free APIs for use in software and web development.

ultimatecourses.com

Readme

Releases

No releases published

Packages

No packages published

Contributors 619

+ 608 contributors

This Lecture's Learning Objectives

Describe at least two types of APIs and data interchange formats

Questions?

Prof. Cody Buntain | @codybuntain | cbuntain@umd.edu
Director, Information Ecosystems Lab