

# Welcome!

---

Webinar:  
Data on the Web: APIs and Libraries

# Speaker Introduction



Christian James

*Web Application Librarian*

Catholic University of America Libraries

- Maintains websites and digital collections
- MLS, University of Maryland, College Park  
MA , History from George Mason University



# Data on the Web



## Application Programming Interfaces (APIs) and Libraries



Christian James, MLS | Sept. 28, 2018  
[jamesc@cua.edu](mailto:jamesc@cua.edu) | @cscottjames

# APIs and Libraries

---

- APIs = Advanced Programming Interfaces

My argument:

- *Coding is becoming a basic competency for information professionals.*
- *Library students should learn how use APIs to take advantage of open data*

# API Examples

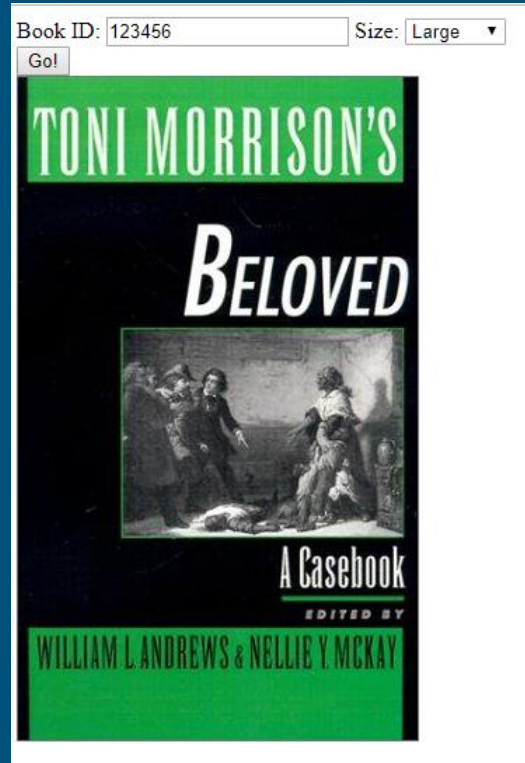
---

- Alma
- LibGuides
- OpenLibrary
- Library of Congress: Chronicling America
- Zotero
- GoodReads
- OpenWeatherMap, Stock data, Marvel Comics
- ...and many, many more
  - <https://github.com/toddmotto/public-apis>



# Simple Applications with JS + APIs

```
1 <!DOCTYPE html>
2 <html>
3 <meta charset="utf-8"/>
4 <body>
5
6 <form id="frm1">
7   Book ID: <input type="text" id="bookNo">
8   Size: <select id="size">
9     <option value="S">Small</option>
10    <option value="M">Medium</option>
11    <option value="L">Large</option>
12  </select>
13 </form>
14 <button onclick="myFunction()">Go!</button>
15
16 <div id="root">
17 </div>
18
19 <script>
20 function myFunction() {
21   var x = document.getElementById("bookNo").value;
22   var elem = document.createElement("img");
23
24   //var bookId = "240727";
25   var size = document.getElementById("size").value;
26   elem.src = 'http://covers.openlibrary.org/b/id/' + x + '-' + size + '.jpg';
27
28   //The appendChild method will **add** to the div
29   //So we need to use innerHTML to empty the div first
30   document.getElementById("root").innerHTML = "";
31   document.getElementById("root").appendChild(elem);
32 }
33
34 </script>
35
36 </body>
37 </html>
```



# Simple Applications:

```
1 <html>
2 <body>
3
4 <div id="root">
5 </div>
6
7 <script>
8   var xhr = new XMLHttpRequest();
9
10  xhr.open("GET", "
11    http://gateway.marvel.com/v1/public/creators/32?ts=1&apikey=
12    [REDACTED], true);
13
14  xhr.send();
15  xhr.onload = function() {
16    var string = xhr.responseText;
17    document.getElementById("root").innerHTML = string;
18  }
19 </script>
20 </html>
```

```
{
  "code": 200,
  "status": "Ok",
  "copyright": "© 2018 MARVEL",
  "attributionText": "Data provided by Marvel. © 2018 MARVEL",
  "attributionHTML": "Data provided by Marvel. © 2018 MARVEL",
  "etag": "1714714027ce0cf0256783b3101d229dc0480f70",
  "data": {
    "offset": 0,
    "limit": 20,
    "total": 1,
    "count": 1,
    "results": [
      {
        "id": 32,
        "firstName": "Steve",
        "middleName": "",
        "lastName": "Ditko",
        "suffix": "",
        "fullName": "Steve Ditko",
        "modified": "2013-01-24T14:41:27-0500",
        "thumbnail": {
          "path": "http://i.annihil.us/u/prod/marvel/i/mg/b/e0/4bc380bd24fe2",
          "extension": "jpg",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/41139",
          "name": "Amazing Fantasy 15: Spider-Man! (2011) #1",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/16214",
          "name": "AMAZING FANTASY OMNIBUS HC (Hardcover)",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6482",
          "name": "The Amazing Spider-Man (1963) #1",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6704",
          "name": "The Amazing Spider-Man (1963) #3",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6879",
          "name": "The Amazing Spider-Man (1963) #6",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6890",
          "name": "The Amazing Spider-Man (1963) #7",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6494",
          "name": "The Amazing Spider-Man (1963) #11",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6527",
          "name": "The Amazing Spider-Man (1963) #14",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6549",
          "name": "The Amazing Spider-Man (1963) #16",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6605",
          "name": "The Amazing Spider-Man (1963) #21",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/6649",
          "name": "The Amazing Spider-Man (1963) #25"
        }
      }
    ]
  }
}
```

# Learning Plan

---

- HTML
- JavaScript
- XML
- JSON
- Ajax

# HTTP Requests

---

Example: <https://postalmuseum.si.edu/visit/calendar.html>

https:// = HTTP protocol; secure encryption

(sub)domain: postalmuseum.si.edu

Path: /visit/

Resource: calendar.html

# DNS Lookup

Computers and other devices communicate using IP addresses to identify each other on the internet.



10.0.0.1  
192.168.1.0  
8.8.8.8

But humans can't remember IP addresses, so they use words.

google.com  
wikipedia.org  
mit.edu  
irs.gov

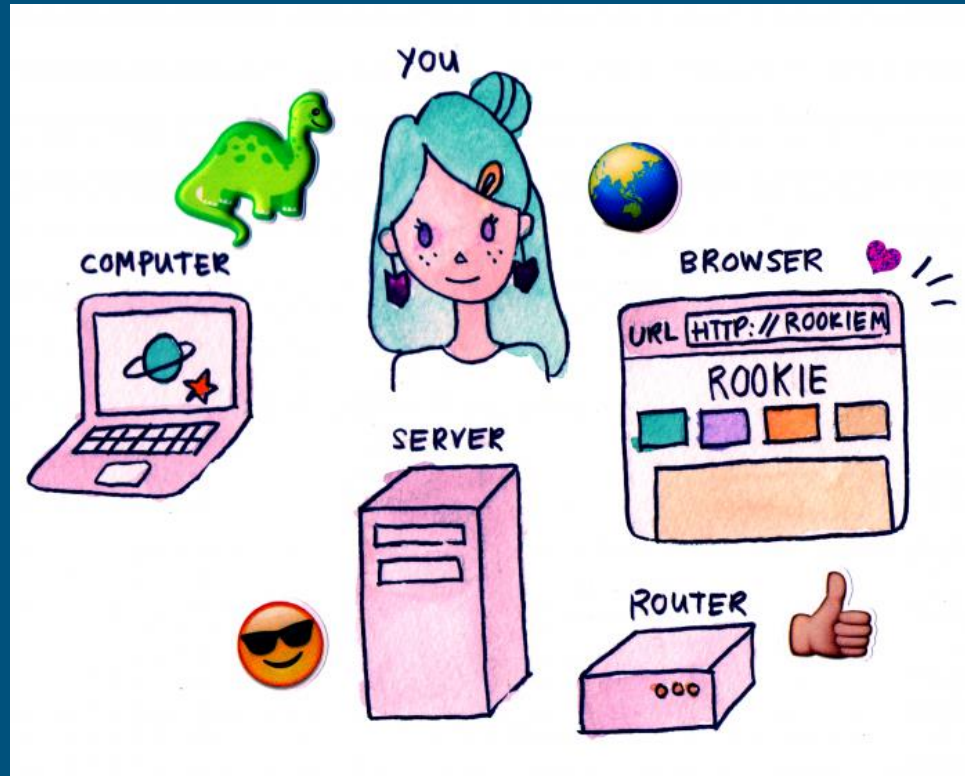


The domain name system (DNS) brings the two together and gets you to your destination.

google.com 216.58.210.142  
mit.edu 95.100.121.86  
wikipedia.org → 91.198.174.192  
irs.gov → 166.123.218.220

# From Web Server to Your Browser: A Journey

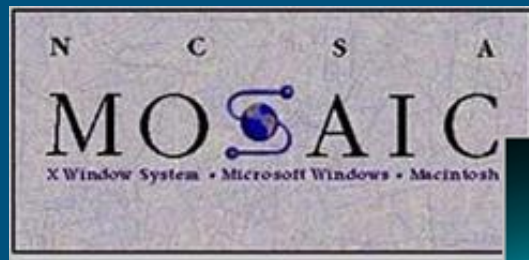
---



# The Browser

---

- Translates HTML and JavaScript into human-readable output
- Limits what can and can't be done for safety
- HTML Standards: World Wide Web Consortium (W3C)
- Browsers can still be manipulated (plugins)



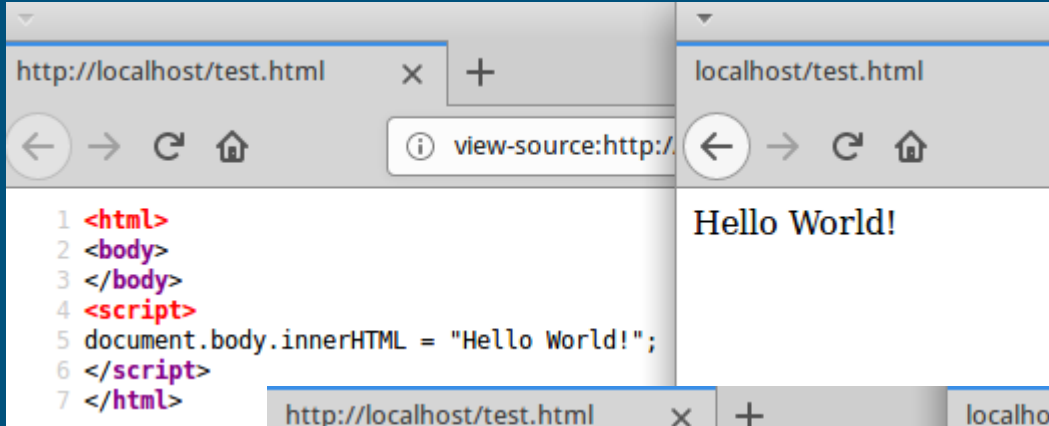
# Client-Side vs. Server-Side

---

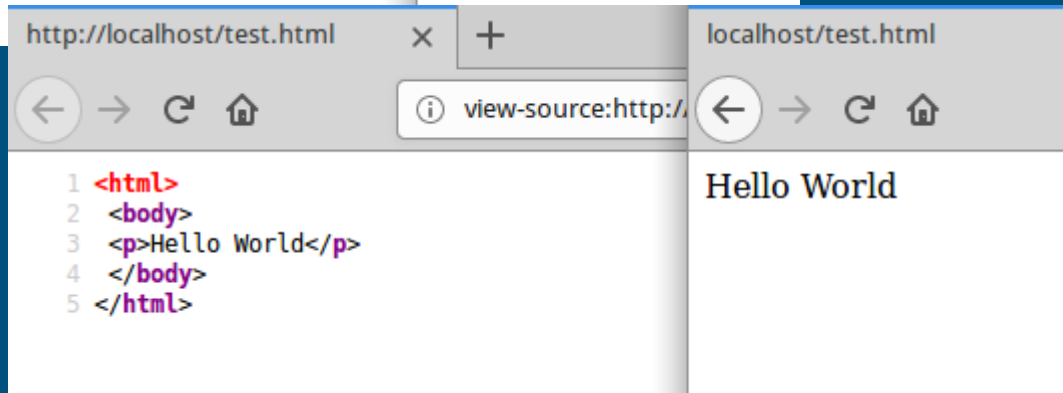
- Client = the connecting computer
- Client-side: code is executed in your BROWSER
  - \*You\* are the “client”
  - Returns unexecuted code; “view-source” doesn’t show the rendered HTML
- Server-side is executed on server
  - “Server” is the computer where the file is hosted
  - View-source returns rendered HTML
- Static vs. “Dynamic” web server



# Client-Side vs. Server-Side: Examples



<- with JavaScript



<- with PHP

# HTML as Object: The DOM

---

- HTML as an object
- Those objects have special properties and methods:
  - Properties:
    - innerHTML
    - Style
    - ...etc.
  - Methods:
    - .getElementsByClassName()
    - .getElementsByTagName()
    - ...etc.
- DOM = “Document Object Model”

# HTML vs. XML

---

- XML = e**X**tensible **M**arkup **L**anguage
- Extensible = extendable/flexible
- Similarities: tag-based system
  - “Semantic” nature of tags
  - Opening/closing
  - Attributes
- HTML: W3C defines the rules
- XML: creator defines the rules
- HTML: great for displaying information
- XML: great for \*describing\* information (metadata)



# JSON

---

```
▼ locations:
  ▼ 0:
    lid: 6712
    name: "Mullen Library"
    category: "library"
    desc: "<p>Notes about Mullen Library go here.</p>"
    url: "http://libraries.cua.edu/"
    contact: ""
    fn: "footnote"
    lat: "38.9337347"
    long: "-76.9954364"
    color: "#000000"
  ▼ weeks:
    ▼ 0:
      ▼ Sunday:
        ▼ times:
          currently_open: false
          status: "open"
          ▼ hours:
            ► 0: {...}
          date: "2018-05-06"
          rendered: "11am - 11:30pm"
      ▼ Monday:
```

# Ajax

---

- Methods:
  - XHR
  - jQuery
  - Fetch

```
function loadDoc() {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            document.getElementById("demo").innerHTML = this.responseText;  
        }  
    };  
    xhttp.open("GET", "ajax_info.txt", true);  
    xhttp.send();  
}
```

[https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)

# API Documentation

GET

/v1/public/comics/{comicId}/stories

Fetches lists of stories filtered by a comic id.

GET

/v1/public/creators

Fetches lists of creators.

### Implementation Notes

Fetches lists of comic creators with optional filters. See notes on individual parameters below.

### Response Class

Model | Model Schema

**CreatorDataWrapper {**

- `code` (int, *optional*): The HTTP status code of the returned result.,
- `status` (string, *optional*): A string description of the call status.,
- `copyright` (string, *optional*): The copyright notice for the returned result.,
- `attributionText` (string, *optional*): The attribution notice for this result. Please display either this notice or the contents of the `attributionHTML` field on all screens which contain data from the Marvel Comics API.,
- `attributionHTML` (string, *optional*): An HTML representation of the attribution notice for this result. Please display either this notice or the contents of the `attributionText` field on all screens which contain data from the Marvel Comics API.,
- `data` (CreatorDataContainer, *optional*): The results returned by the call.,
- `etag` (string, *optional*): A digest value of the content returned by the call.

**}**

**CreatorDataContainer {**

- `offset` (int, *optional*): The requested offset (number of skipped results) of the call.,
- `limit` (int, *optional*): The requested result limit.,
- `total` (int, *optional*): The total number of resources available given the current filter set.,
- `count` (int, *optional*): The total number of results returned by this call.,
- `results` (Array[Creator], *optional*): The list of creators returned by the call.

**}**

[https://developer.marvel.com/docs#!/public/getCreatorCollection\\_get\\_12](https://developer.marvel.com/docs#!/public/getCreatorCollection_get_12)

# API Data

---

```
{
  "code": 200,
  "status": "Ok",
  "copyright": "© 2018 MARVEL",
  "attributionText": "Data provided by Marvel. © 2018 MARVEL",
  "attributionHTML": "Data provided by Marvel. © 2018 MARVEL",
  "etag": "1714714027ce0cf0256783b3101d229dc0480f70",
  "data": {
    "offset": 0,
    "limit": 20,
    "total": 1,
    "count": 1,
    "results": [
      {
        "id": 32,
        "firstName": "Steve",
        "middleName": "",
        "lastName": "Ditko",
        "suffix": "",
        "fullName": "Steve Ditko",
        "modified": "2013-01-24T14:41:27-0500",
        "thumbnail": {
          "path": "http://i.annihil.us/u/prod/marvel/i/mg/b/e0/4bc380bd24fe2",
          "extension": "jpg",
          "resourceURI": "http://gateway.marvel.com/v1/public/comics/60533",
          "available": 467,
          "collectionURI": "http://gateway.marvel.com/v1/public/creators/32/comics",
          "items": [
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/60533",
              "name": "Alpha Flight by John Byrne Omnibus (Hardcover)"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/16926",
              "name": "Amazing Fantasy (1962) #15"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/41139",
              "name": "Amazing Fantasy 15: Spider-Man! (2011) #1"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/16214",
              "name": "AMAZING FANTASY OMNIBUS HC (Hardcover)"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6482",
              "name": "The Amazing Spider-Man (1963) #1"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6704",
              "name": "The Amazing Spider-Man (1963) #3"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6879",
              "name": "The Amazing Spider-Man (1963) #6"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6890",
              "name": "The Amazing Spider-Man (1963) #7"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6494",
              "name": "The Amazing Spider-Man (1963) #11"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6527",
              "name": "The Amazing Spider-Man (1963) #14"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6549",
              "name": "The Amazing Spider-Man (1963) #16"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6605",
              "name": "The Amazing Spider-Man (1963) #21"
            },
            {
              "resourceURI": "http://gateway.marvel.com/v1/public/comics/6649",
              "name": "The Amazing Spider-Man (1963) #25"
            }
          ]
        }
      }
    ]
  }
}
```

# API Data, cont.

```
1 <html>
2 <body>
3
4 <div id="root">
5 </div>
6
7 <script>
8   var xhr = new XMLHttpRequest();
9
10  xhr.open("GET", "
11  http://gateway.marvel.com/v1/public/creators/32?ts=1&apikey=[REDACTED]
12  ", true);
13  xhr.send();
14  xhr.onload = function() {
15    var string = xhr.responseText;
16    var code = JSON.parse(string).data.results[0].comics.items;
17    var i = "";
18    for (i = 0; i < code.length; i++) {
19      document.getElementById("root").innerHTML += code[i].name + "<br />";
20    }
21  }
22 </script>
23 </html>
```

Alpha Flight by John Byrne Omnibus (Hardcover)  
Amazing Fantasy (1962) #15  
Amazing Fantasy 15: Spider-Man! (2011) #1  
AMAZING FANTASY OMNIBUS HC (Hardcover)  
The Amazing Spider-Man (1963) #1  
The Amazing Spider-Man (1963) #3  
The Amazing Spider-Man (1963) #6  
The Amazing Spider-Man (1963) #7  
The Amazing Spider-Man (1963) #11  
The Amazing Spider-Man (1963) #14  
The Amazing Spider-Man (1963) #16  
The Amazing Spider-Man (1963) #21  
The Amazing Spider-Man (1963) #25  
The Amazing Spider-Man (1963) #26  
The Amazing Spider-Man (1963) #27  
The Amazing Spider-Man (1963) #30  
The Amazing Spider-Man (1963) #31  
The Amazing Spider-Man (1963) #32  
The Amazing Spider-Man (1963) #33  
The Amazing Spider-Man (1963) #35

# Using APIs: Scenarios

---

- Real-world projects:
  - Load OCLC metadata for cataloging projects
  - Manage library hours on website
  - Display the availability of a book based on catalog item status
- Speculative student projects:
  - Display today's weather
  - Status of a bus line on a nearby bus stop
  - Information about a local sports team (games, stats, etc.)
  - Information about a game or video game for a school media center or youth services division

# Disadvantages and Advantages

---

- Disadvantages
  - Cross-origin policies: CORS
  - Exposing API Keys
  - Extra code
- Advantages
  - No extra downloads, installation

# Further Resources & Study

---


- Resources:
  - Working with Data on the Web: <https://www.lynda.com/CSS-tutorials/Working-Data-Web/133326-2.html>
  - Studio Ghibli case study: <https://www.taniarascia.com/how-to-connect-to-an-api-with-javascript/>
- To learn next:
  - Web Scraping
  - IIIF
  - XPath
  - XSLT



# Thank You!



Christian James, MLS | Sept. 28, 2018  
[jamesc@cua.edu](mailto:jamesc@cua.edu) | @cscottjames



# Q&A

# Thank you!

Connect with Us:

[SJSU ASIS&T Website](#)



[Facebook](#)



[Twitter](#)



[YouTube](#)



[Mailing List](#)