

Week 02

Post-A(P)I Age

Web Data Science

Brian Keegan, Ph.D.

Assistant Professor, Information Science

brian.keegan@colorado.edu



University of Colorado
Boulder

Course Overview

Module	Week	Dates	Topics
<i>Fundamentals</i>	1	Aug 27; Aug 29	Law & ethics
	2	Sep 3; Sep 5	Post-A(P)I Age
	3	Sep 10; Sep 12	XML & JSON
	4	Sep 17; Sep 19	IP & HTTP
Module Assignment 1 due September 23			
<i>Documents</i>	5	Sep 24; Sep 26	Static web pages
	6	Oct 1; Oct 3	Archived web pages
	7	Oct 8; Oct 10	Dynamic web pages
	8	Oct 15; Oct 17	PDFs
Module Assignment 2 due October 21			
<i>APIs</i>	9	Oct 22; Oct 24	Wikipedia
	10	Oct 29; Oct 31	Government
	11	Nov 5; Nov 7	Social
	12	Nov 12; Nov 14	AI
	13	Nov 19; Nov 21	Automation
Module Assignment 3 due December 2			
	14	Nov 26; Nov 28	No Class: Fall Break
<i>Final projects</i>	15	Dec 3; Dec 5	Final projects
	16	Dec 10; Dec 12	

Tuesday

Agenda

- 00:00 – 00:05 → Daily Note check-in
- 00:05 – 00:20 → Architecture of the web
- 00:20 – 00:40 → Browser developer tools
- 00:40 – 01:00 → Running a (local) web server
- 01:00 – 01:15 → Final Project, Next Class

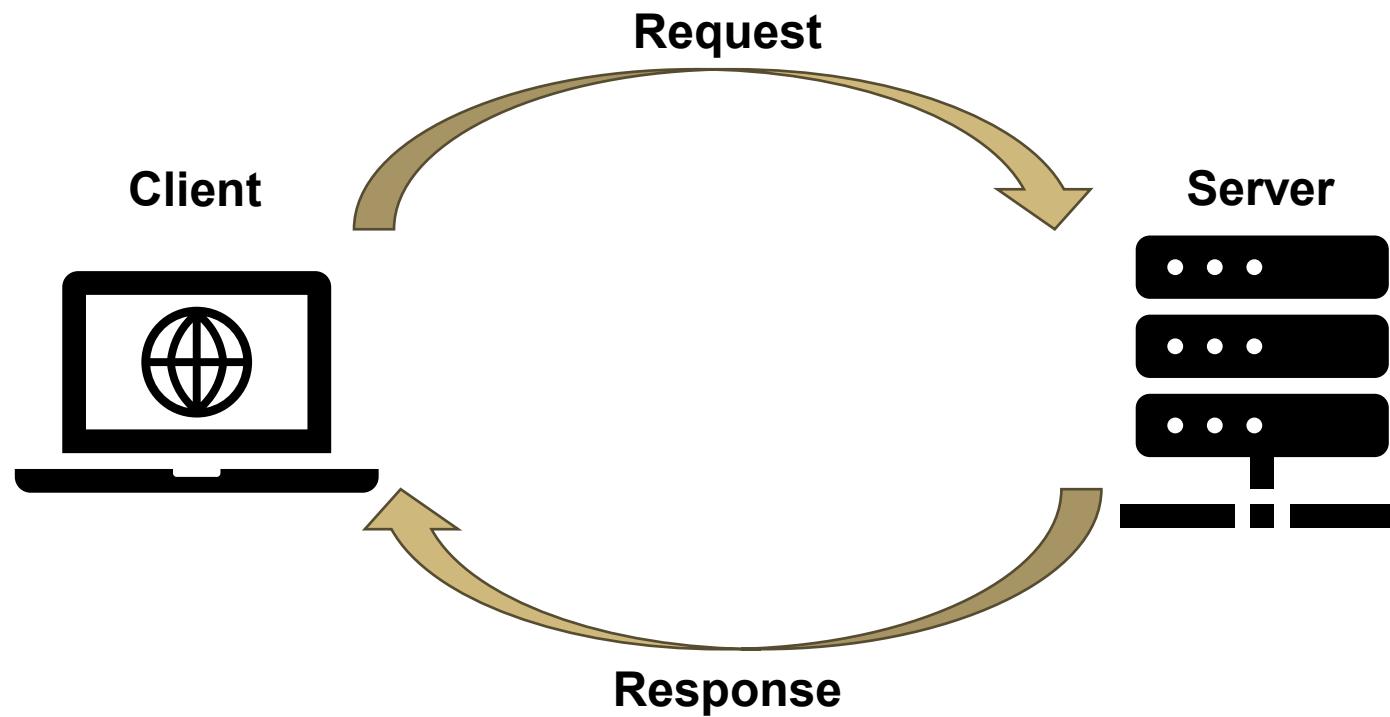
Daily Note check-in

- Any pets?
 - Do worms for composting count?
- What stops people from header spoofing?
 - Ethics and lots of other engineering to prevent undesirable GETs
- Buffs score tonight?
 - 28-10
- Favorite restaurant in Colorado?
 - Blackbelly/Santo (\$\$\$), Avery (🍷), Tierra y Fuego (\$), Gelato Boy (🍦)

Architecture of the web

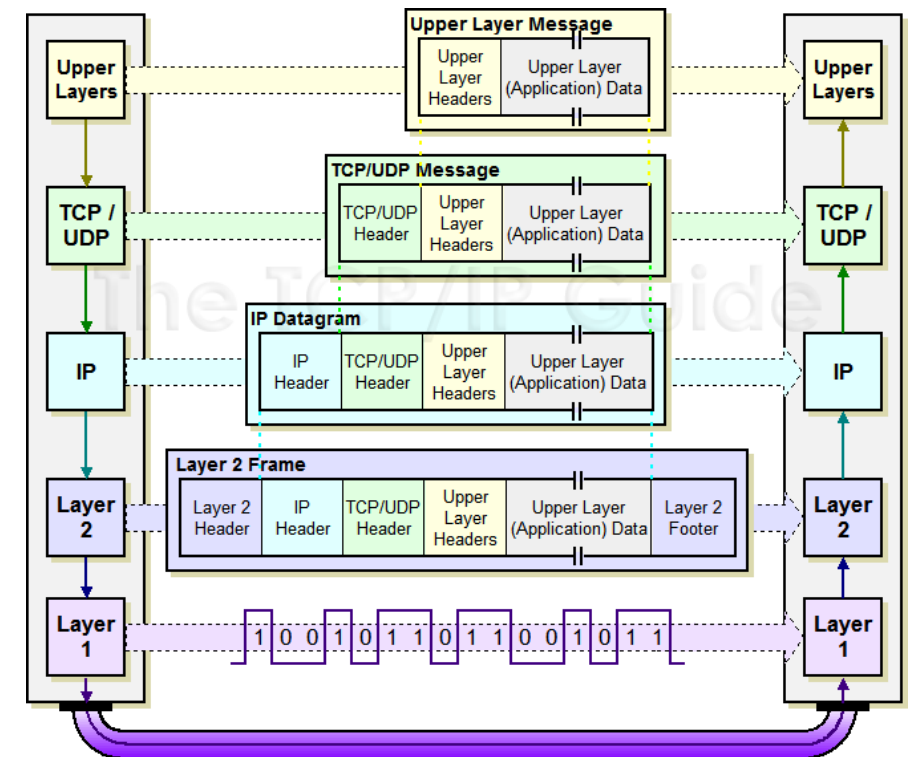
- A whole lot happening behind the scenes of loading a webpage!
- Client-server model
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- Domain Name System (DNS)
- Hypertext Transfer Protocol (HTTP)
- Files

Client-server model



TCP/IP

- “Transportation” of the web
- TCP/IP controls how data moves from place-to-place
- Created back in the 1970s to support DoD mission of ensuring network robustness under attack
- Works in layers
 - Application data is broken down into packets of information
 - Packet headers with instructions about where to send them
 - Packets move along literal pipes to their destination
 - Packets are re-assembled at destination into application data
- If you’re debugging TCP/IP packets, something is *really* wrong



<http://www.tcpipguide.com/free/diagrams/ipencap.png>

Domain Name System

- “Address book” of the web
- IP addresses are locations of devices on a network – what is yours?

```
> ipconfig
```
- Every device has a unique address
 - IPv4 has $2^{32} = 4.3e9$ possible addresses, these would exhaust around 2010 → 151.101.130.133
 - IPv6 has $2^{128} = 3.4e38$ possible addresses → eight groups of four hexadecimal digits
- Domain Name System (DNS) is the address book that maps servers to IP addresses
 - <https://www.nslookup.io/>
- Domain names are maintained by ICANN, DNS servers are fundamental infrastructure
 - If you’re debugging DNS issues (for someone else’s website), something is *really* wrong

Hypertext Transfer Protocol

- “Language” of the web
- Application-layer language for exchanging documents
- Clients (browsers) establish a connection
 - Send requests like GET or POST
 - Servers send responses
 - For a GET, server processes and sends requested data
 - Client receives data (files)
- If you’re debugging HTTP requests/responses, something is *really* wrong

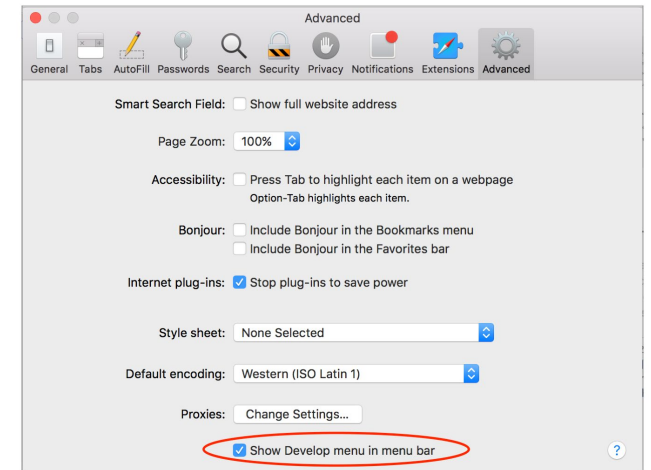
```
GET / HTTP/1.1  
Host: developer.mozilla.org  
Accept-Language: en
```

Files

- Code
 - Hypertext markup language (HTML) → Describes content of a web page
 - Cascading Style Sheets (CSS) → Style and layout of a web page
 - JavaScript (JS) → Providing interactivity for a web page
- Assets
 - Images (PNG, JPG, SVG, GIF)
 - Music
 - Video
 - Portable Document Format (PDF)
- If you're debugging HTML, CS, and JS files, you're a web data scientist!

Browser developer tools

- Windows: CTRL + Shift + I or F12
- Mac: ⌘ + ⌥ + I
- Right click and select Inspect
- In Chrome:
 - More tools > Developer tools
- In Safari:
 - Safari > Settings > Advanced > “Show features for web developers”
 - Develop > Show Web Inspector
- In Firefox:
 - Hamburger > More tools > Web Developer Tools



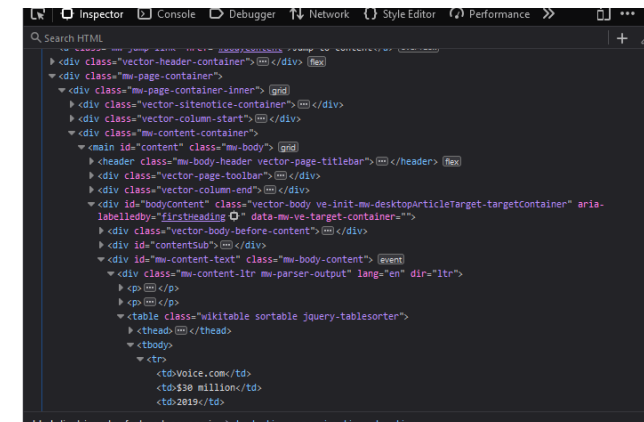
Developer tools: Inspector

- Most modern websites are full of tracking garbage → Start with Wikipedia instead
- Find a (simple) article on Wikipedia and open your browser's Developer Tools
- What is the source code for this page? Go to the Inspector tab to see the HTML of the page
- Highlighting tags should highlight what the tag (and its descendants) define
- Open tags underneath until you can't anymore



The screenshot shows the Wikipedia article titled "List of most expensive domain names". The article text states that this is a list of domain names that sold for \$3 million USD or more, limited to pure domain name and cash-only sales. Below the text is a table with the following data:

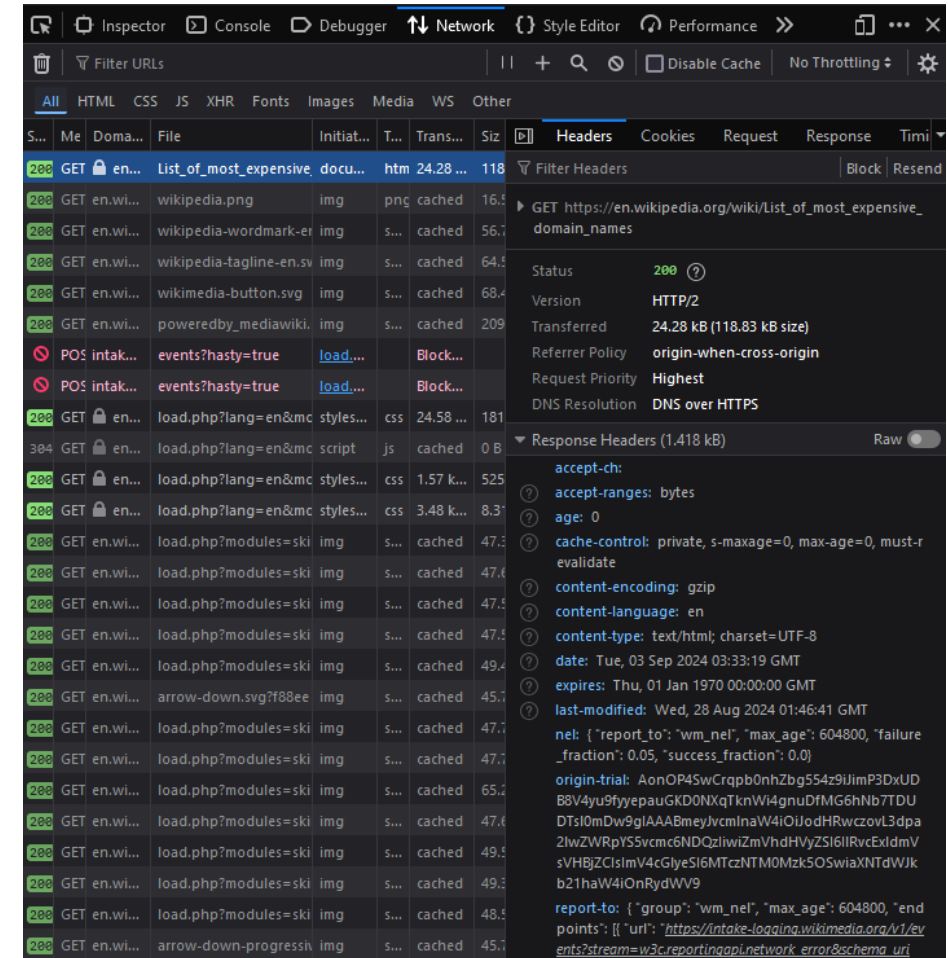
Domain	Price	Sale date	Ref
Voice.com	\$30 million	2019	[1]
360.com	\$17 million	2015	[2]
NFTs.com	\$15 million	2022	[3]
Sex.com	\$13 million	2010	[4]
Hotels.com	\$11 million	2001	[5]
Tesla.com	\$11 million	2014	[6]



The screenshot shows the browser's Developer Tools Inspector. The HTML structure is displayed, showing the document's root element and various nested containers. The 'main id="content"' element is selected, and its children are visible, including the header, toolbar, and the main content area. The main content area contains a table with the same data as the one in the Wikipedia article screenshot.

Developer tools: Network

- Examine all the requests made between your browser and other servers to provide the data to render the web page
 - How many? What kinds? What domains?
- Filter by file type (HTML, CSS, JS, images, etc.)
- Click on one to see headers associated with the request and response
- Headers provide more structured details on what kinds of information to include and how to format it
- What's hiding inside the cookie header?
- What is the “User-Agent” header?
- What is the “Referer” header?



Running a local web server

- Download “index.html” from Canvas and put it into your class directory
- Open a Terminal window, navigate to your class directory
- In Terminal, run:

```
> python -m http.server 8088
```
- In a web browser, go to localhost:8088
- Open index.html file in a text editor, change something in the file, save, and reload the browser

Exercises for Thursday

- Use Developer Tools on a complex website (Instagram, Amazon, NYTimes, *etc.*)
 - How many requests? What kinds of domains? How much data?
- Edit the content of a tag in Developer Tools to spoof a web page
- Copy the content of a webpage to a text editor and count how often an interesting value appears
- Make a slightly more complex index.html
 - https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics
 - Add content and styles
 - Include images or links to other pages

Readings for Thursday

- Freelon (2018) – Computational Research in the Post-API Age
 - Perspective from 2018 after Facebook closed off an API on which researchers depended
- Bruns (2019) – After the ‘APIcalypse’: social media fight against critical scholarly research
 - Loss of API access is diminishing ability to conduct independent critical research
- Puschmann (2019) – An end to the wild west of social media research
 - Lots of abusive research practices beforehand, maybe restrictions + partnerships are defensible?

Next Class

- Try out exercises and come ready to share
- Discuss readings
- Daily note: <https://bit.ly/info4871f24note>
- Today's number is...

Thursday

Agenda

- 00:00 – 00:05 → Daily Note check-in
- 00:05 – 00:40 → Review exercises
- 00:40 – 01:10 → Discussing readings
- 01:10 – 01:15 → Wrap up and Daily Note

Daily Note