University of California, Berkeley
Master of Information and Data Science (MIDS)
W205 – Fundamentals of Data Engineering

Week 12 – Web APIs, Part I

Agenda for Today’s Class

- Attendance and Participation
- Announcements
- Schedule and Due Dates
- Work / Life / School Balance
- Asynch High Level Review in a Nutshell
- Breakouts
- Summary

Attendance and Participation

Please record your attendance and participation for today’s class:

GitHub => ucb_mids_w205_repo => README.md => Attendance and Participation

Announcements

- Upcoming holidays and/or breaks
- Makeup classes for holidays
- Upcoming events
- Student evaluations
- Etc.

Schedule and Due Dates

Take a quick look at the next couple of weeks’ due dates:

GitHub => ucb_mids_w205_repo => README.md => Schedule and Due Dates

Work / Life / School Balance

Open Discussion

Student feedback

- About 5 minutes
- How are things going related to work / life / school balance?
- How is w205 going? Difficulty? Time?
- Impact of any natural and/or man-made disasters
- Etc.

Asynch High Level Review in a Nutshell

Each week we will spend about 15 minutes reviewing the most important high level concepts from the asynch

Web Servers are Stateless by Default

- Stateless protocol
  - Each packet of information sent by the client to the server is meaningful in isolation
  - Allows scale up
- HTTP / HTTPS is a stateless protocol
- Web Servers use HTTP / HTTPS, therefore they are stateless
  - Web Servers scale up very well!

Creating a Stateful Web Server

- Web server creates a unique SID (session ID) to save state for each user
- Without login
  - Web server creates a client side cookie and sets it to the SID
  - Tracking cookies can live for days, weeks, months
- With login
  - When user logs in, web server creates a SID and sets client side cookie
  - Subsequent requests use the SID to retrieve and update state
  - When user logs out, web server destroys SID

Web Server Scale Up

- Static content is pushed using CDN
- Public internet is connected to reverse proxies
- Reverse proxies are hot railed to web servers
- Web servers are hot railed to application servers
- Application servers are hot railed to a transactional database
- Semi-static pieces of the transactional database layers can be pushed out to application servers using the big data immutable model

Screen Scraping

- A computer program visits a website in the same manner as a human at the keyboard would.
- The website thinks that the computer program is a human.
- Used when no API provided

Screen Scraping Issues

- Legal issues
- Ethical issues
- Violation of terms of service
- Websites block screen scrapers
  - Robot tests, change HTML frequently, data in images, etc.
- HTML parsing – inexact, may change without notice
- Client-side scripts have to be run before output can be parsed
- Data in images needs some sort of OCR

Downloads

- For data that is commonly requested, it is often more efficient for a website to provide the data in download file(s)
- Downloads use HTTP / HTTPS
  - GET for common dataset
  - POST for custom dataset
  - Response
    - Text, UTF-8
    - Binary, Base64 encoded (uuencoded)
  - Files nested in Zip files may have issues with binary files not in UTF-8

Breakouts

GitHub => ucb_mids_w205_repo => breakouts

(time permitting, we may not get to all of them)

Summary

Instructor will give a brief (about 2 minute) summary of today’s class.