Sessions and Cookies

Review: Dynamic Pages

- Servers can **dynamically generate** content to send to a client
  - Backend software helps create HTML on the fly
  - **Server-side dynamic** pages require full reloads on the client
  - **Client-side dynamic** pages use JavaScript to modify the DOM without interacting with the server

- We can use **Flask** for Python to create various **routes** that allow dynamic creation of content
  - We use Flask by defining methods and **decorating** them with @app.route() calls to create **routes**
  - Flask's structure *abstracts away* the filesystem – the **path** of a URL no longer refers to a **file** on the server, but encodes the *semantics of a request*

Review: JavaScript

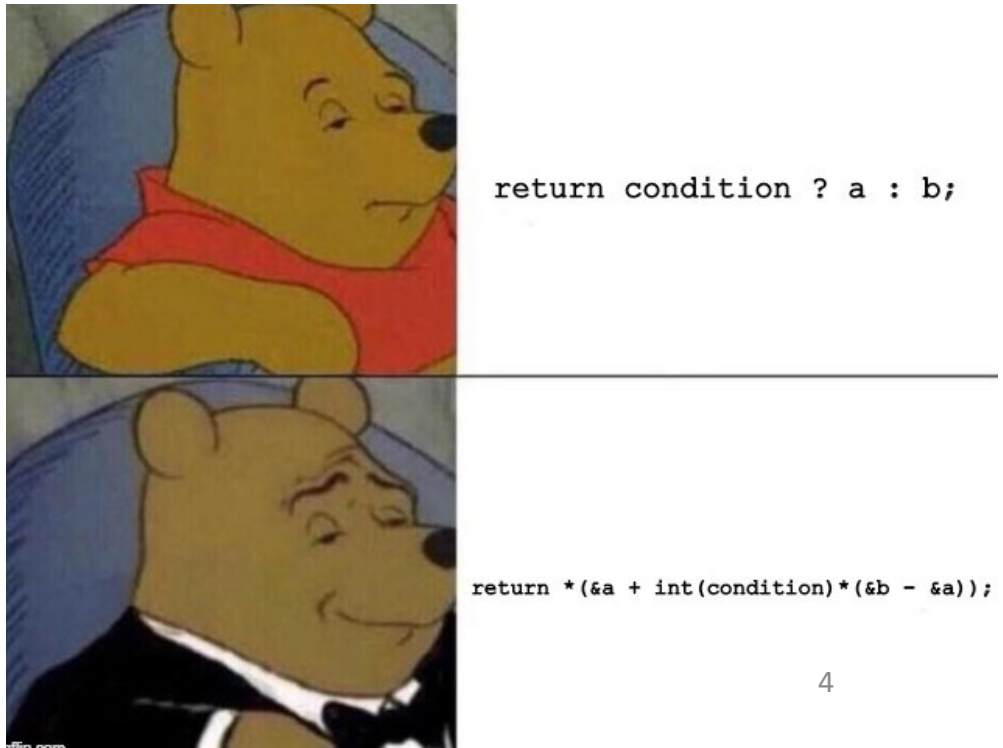
- **JavaScript** is code embedded within HTML
  - JS lets you modify parts of the DOM tree
- JS is based around **asynchronous events**
  - Write a function that runs when:
    - the user clicks a mouse button
    - the user moves the mouse
    - the user presses a key
    - the page finishes loading
- You like JavaScript

```
<html>
<head>
<body>
<button id="cool">Click me</button>
<script>
 document.getElementByID("cool").onclick =
   function () { alert('hello'); };
</script>
</body></html>
```

Review: Python Decorators

- A **Decorator** is *syntactic sugar* in Python
- “@” prefix before something is a decorator
  - Decorators basically wrap functions in other functions

- Click library
  - Decorate a function with command line inputs that it can accept
- Flask library
  - Decorate functions that represent URL endpoints that get served



Review: Decorators

```
def add_message(fn):
      def inner():
            print("Calling...")
            fn()
      return inner

@add_message
def f():
      print("f ran")

f()                # “Calling…” followed by “f ran”
f()
```

One-Slide Summary: Sessions

- We use **sessions** to help maintain *state* between requests
  - Recall: HTTP is stateless, so we build sessions on top of the server and client to hack around this limitation
- **Sessions** are data stored on the *server*
  - Usually, some sort of map or dictionary structure; (key,value) pairs
  - e.g., uniqname=kjleach, cart="1,2,3"
- **Cookies** are (key,value) pairs that are stored on the *client*
  - **Cookies** can help the server figure out which **session** is appropriate for a response

Sessions

- Sessions are used for
  - Knowing which user is logged in
  - Tracking the pages a user accesses
  - Remembering things in a shopping cart
  - **Web browser level**

- Not used for
  - Knowing which computer to send a response to
  - **Network level**

Why sessions are necessary

- Web requests use HTTP
- HTTP is "stateless": doesn't keep track of user who sent a request
  - (recall from Flask: there’s nothing built in that tracks an individual…)

- To allow for sessions (e.g. logging in), need a layer on top of HTTP
  - In Flask: your script will manage variables that help track individual users!

Sessions

- A session is a single "interaction" between the site and user
  - Precise definition depends on application

- Example: Facebook or Gmail login
- Example: Amazon cart
  - Even when you’re not logged in

Amazon cart: example

Morton Salt, Iodized, 26 Ounce

by Morton

393 ratings | 4 answered questions

Price: $0.95 ($0.04 / Ounce) prime

- ALL PURPOSE - This all-purpose salt features uniformly shaped crystals making it the perfect SALT when precise measurements are critical
- GREAT FOR EVERYDAY USE - This is an Iodized salt, perfect for everything from cooking and baking to filling table salt shakers
- IODIZED - This salt supplies iodine, a necessary nutrient for proper thyroid functions
- VERSATILE - Meets various operational needs - both front and back of house
- EASY TO USE - Simplify your cooking needs in the kitchen with Morton all-purpose table salt

$0.95

prime

FREE Delivery by **Sunday, July 12** for Prime members

1
2
3
9001

on July 10, 2020.

w.

Added to Cart

Cart subtotal (9001 items): **$0.95**

This is a gift

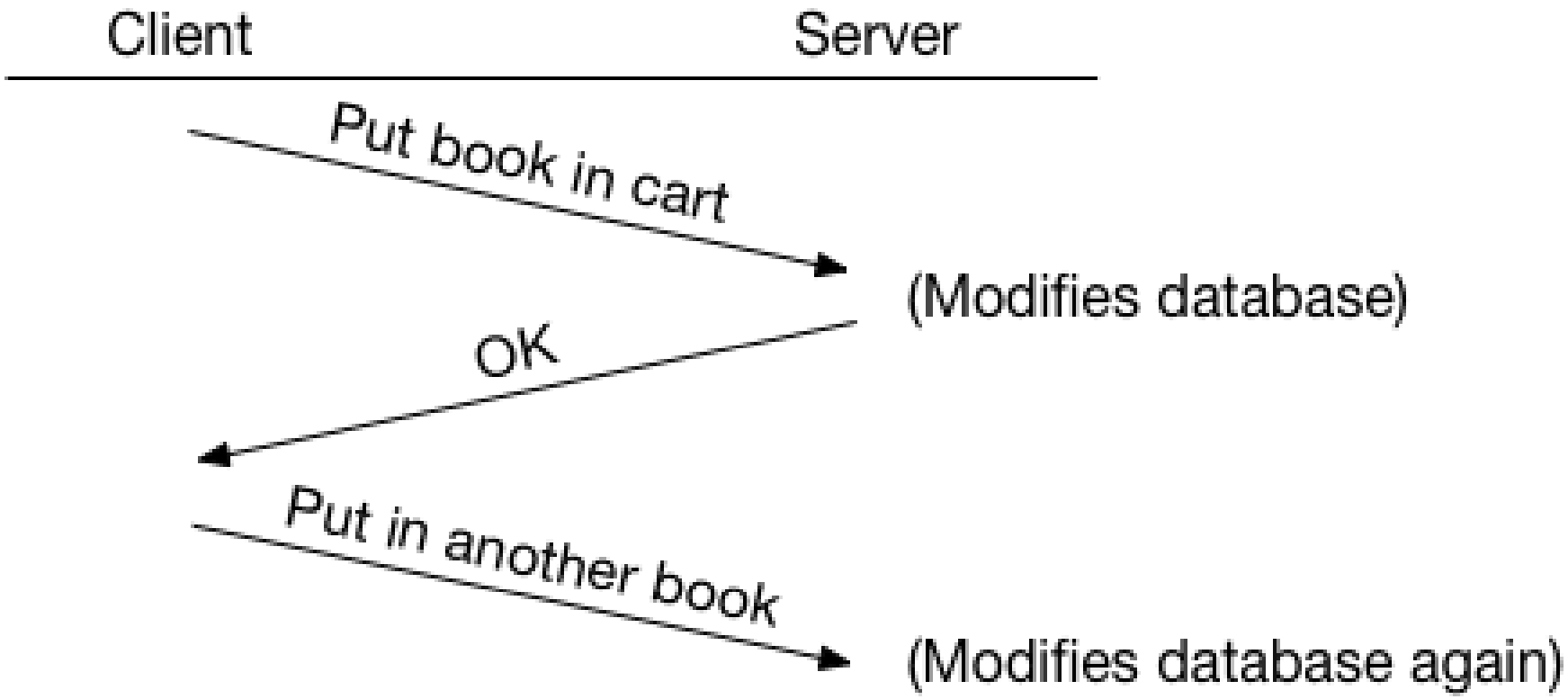
Cart

Proceed to checkout (9001 items)

Thought Question

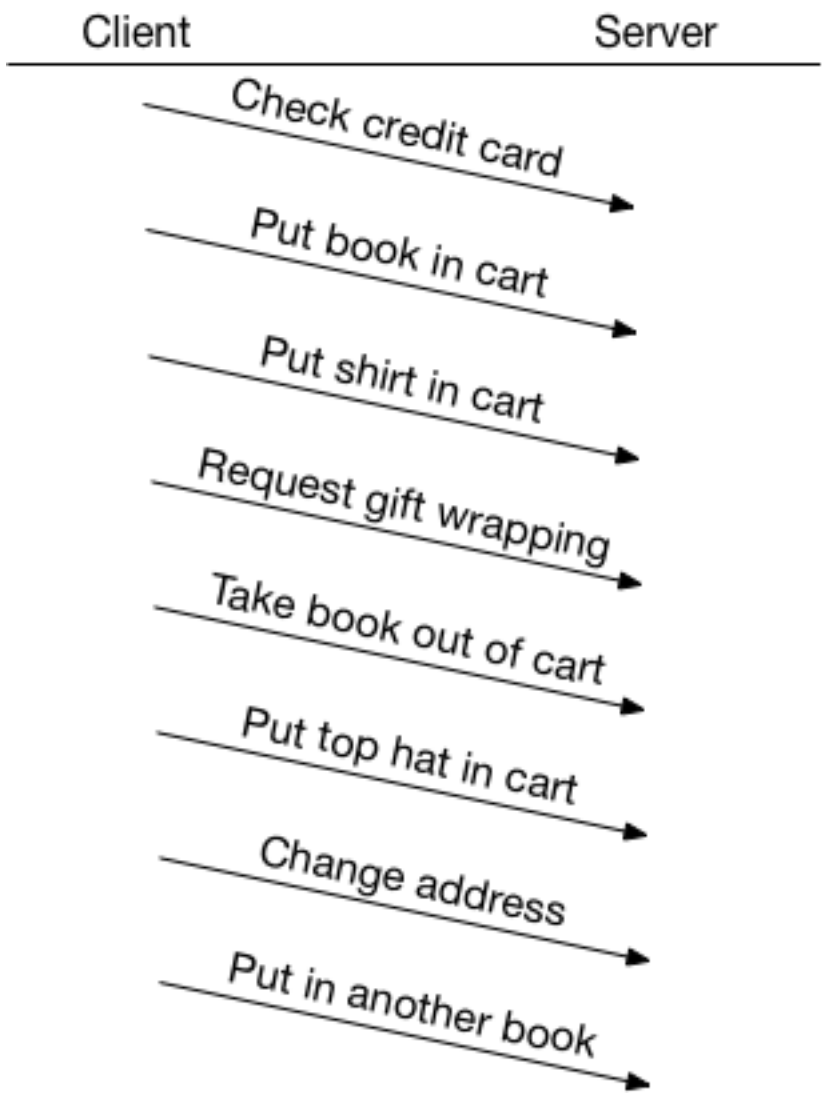
- If I open an " incognito mode" window in Chrome:
  - My Amazon shopping cart is empty in the incognito window
  - I can then log in as a second user in incognito
  - I'm still logged in as the original user in the non-incognito window
- How many sessions are active in this situation?
- Can you think of another situation where multiple sessions on the same computer is useful?

Session perspectives - client



Session perspectives: server

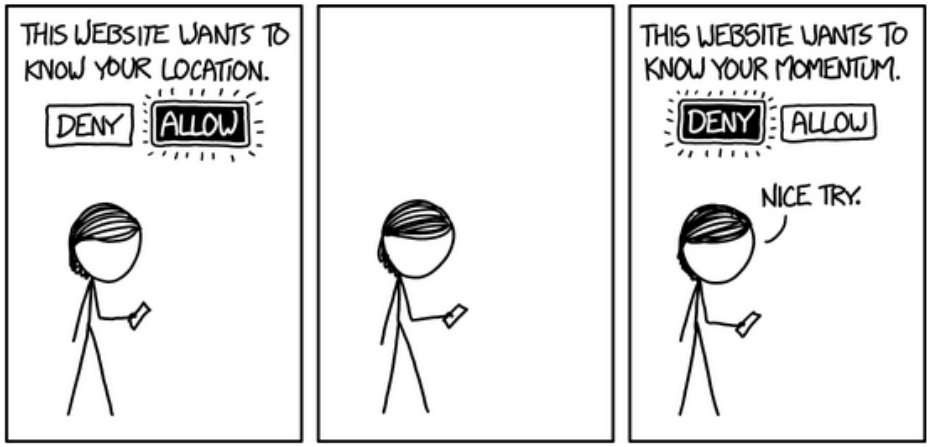
- A server must track many sessions at once
- How can it tell the difference between clients?



Sessions and stateless HTTP

- HTTP is stateless, so we want to use a "session protocol" on top of it
  - JK there's no such thing as a “session protocol” #lol
- Implemented at application layer instead
  - State maintained in session variables
  - Data stored in one request can be accessed by later request
    - Caution: Flask is not durable – if you manage sessions with variables only, they’ll go away when you stop the script.
      - Use a DB instead
- Application layer sessions are one reason to use a web framework
  - Flask, Django, Ruby-on-rails, etc.

Implementing sessions



- Problems to solve:
  - Creating and destroying sessions
    - example: logging in/out
  - Linking sessions to users
    - I can be logged into Gmail on both my laptop and phone at same time
  - Linking HTTP requests to sessions
    - Keep track of individual browser tabs that might connect to the same website

When to create a session?

- Depends on the application
- Amazon?
  - When you visit the page
  - Needed for a shopping cart
- Gmail?
  - When you log in
  - Needed so that it shows the right mail to the right person
- Projects 2 and 3?
  - Wouldn’t it be nice to store whether someone logged in and not prompt on every request?

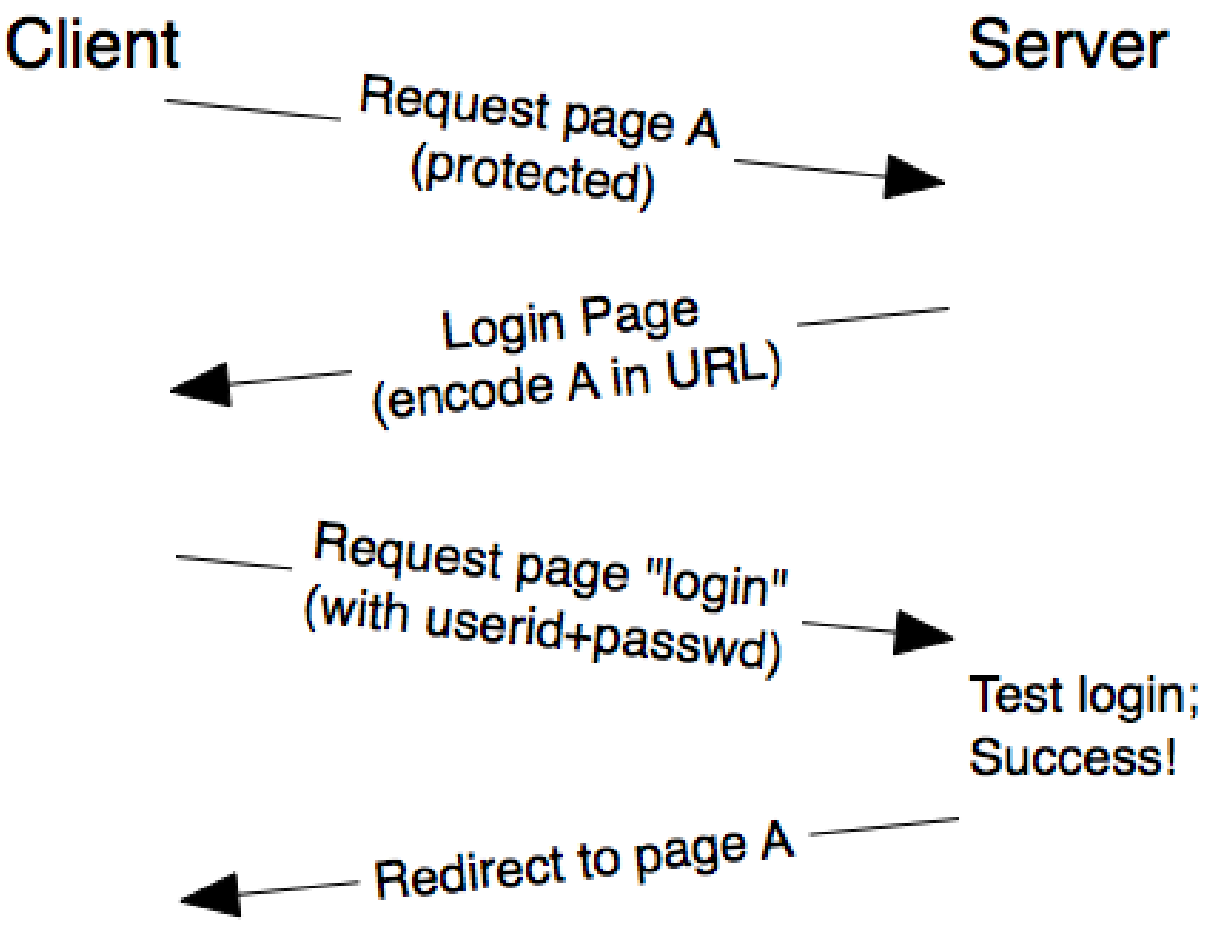
How to store session data?

- Session data storage is up to the server
- Best practice: store a small amount of data identify the session
  - Username, session ID, etc.
- Use session ID or username to do a database lookup
  - Shopping cart content, news feed items, etc.

When to close a session?

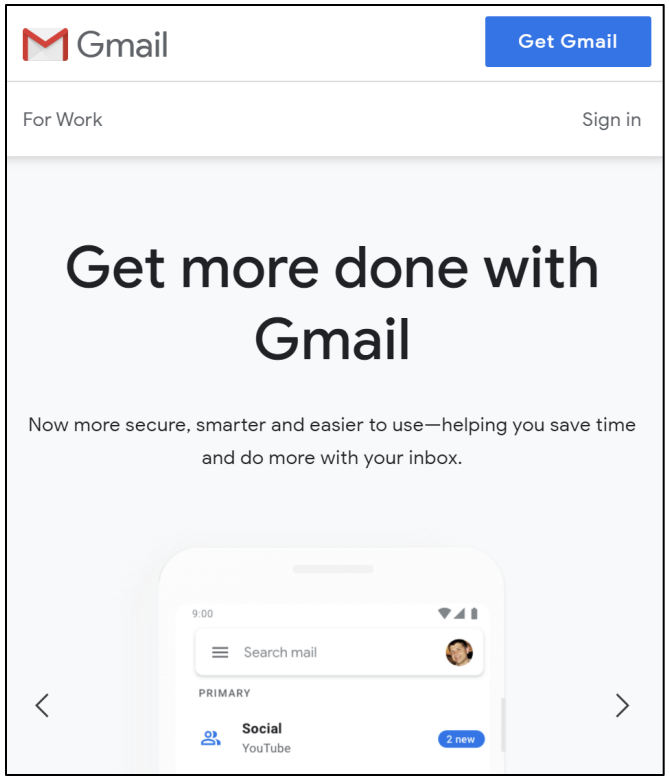
- We can't rely on logout
- Timeouts needed for almost all apps
  - When should the online game be reset?
  - When should Google forget your search?
  - When has your cart been abandoned?
  - When have you started searching for a different flight?
- Timeout from first request or most recent?

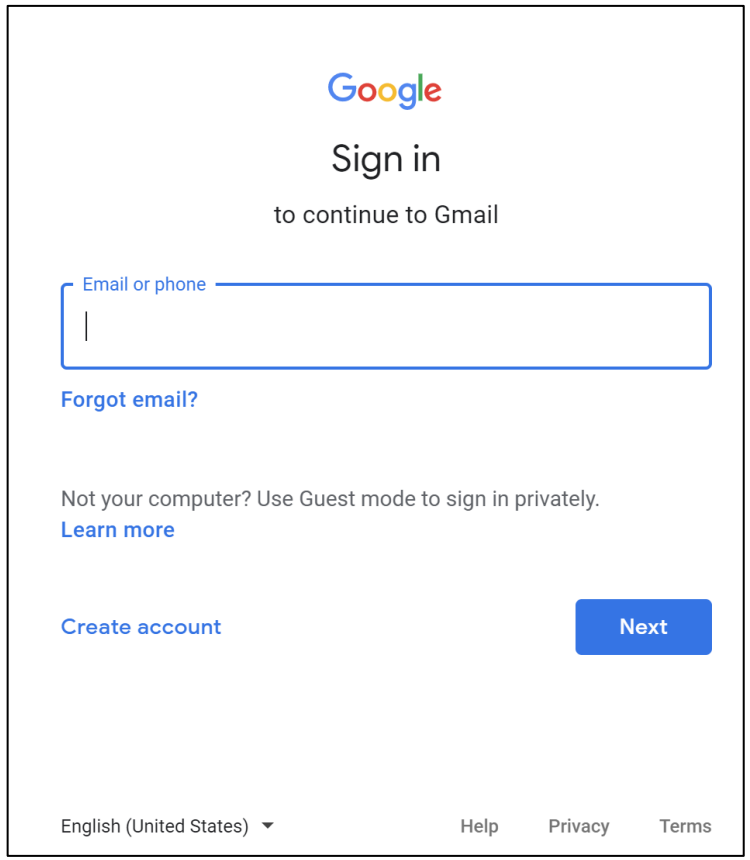
How to link sessions to users

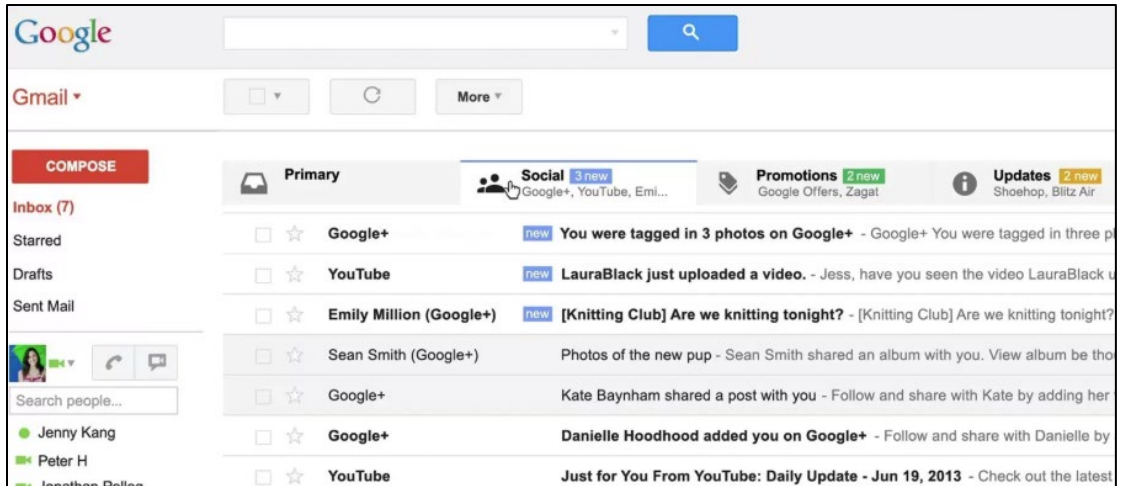


How to link sessions to users

1. Client requests https://mail.google.com
2. Server responds with redirect to https://accounts.google.com/signin/v2/identifier?**continue=https%3A%2F%2Fmail.google.com%2Fmail%2F**
   - Recall URL escaping: `%3A` == `:`, `%2F` == `/`
3. Client sends request with username and password
4. Server tests and responds with redirect to https://mail.google.com


Gmail
Get Gmail
For Work
Sign in
Get more done with Gmail
Now more secure, smarter and easier to use—helping you save time and do more with your inbox.
9:00
Search mail
PRIMARY
Social
YouTube
2 new

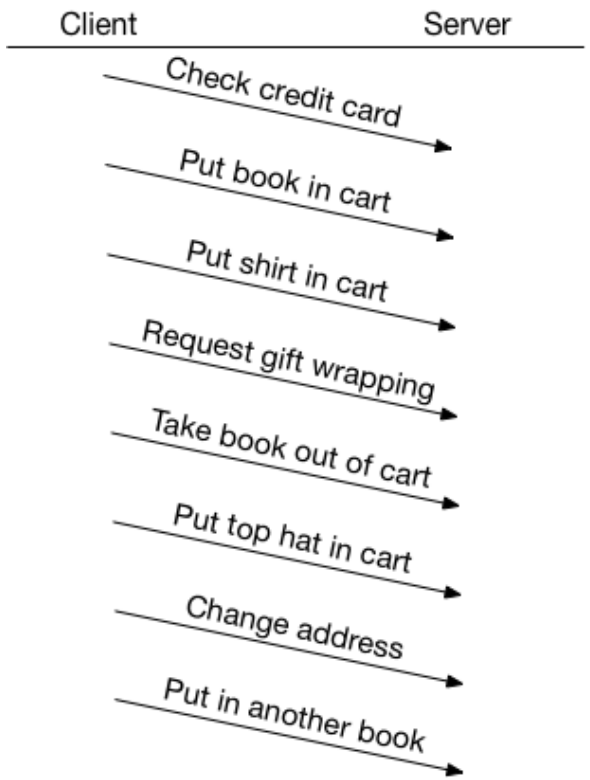

Google
Sign in
to continue to Gmail
Email or phone
Forgot email?
Not your computer? Use Guest mode to sign in privately.
Learn more
Create account
Next
English (United States)
Help
Privacy
Terms

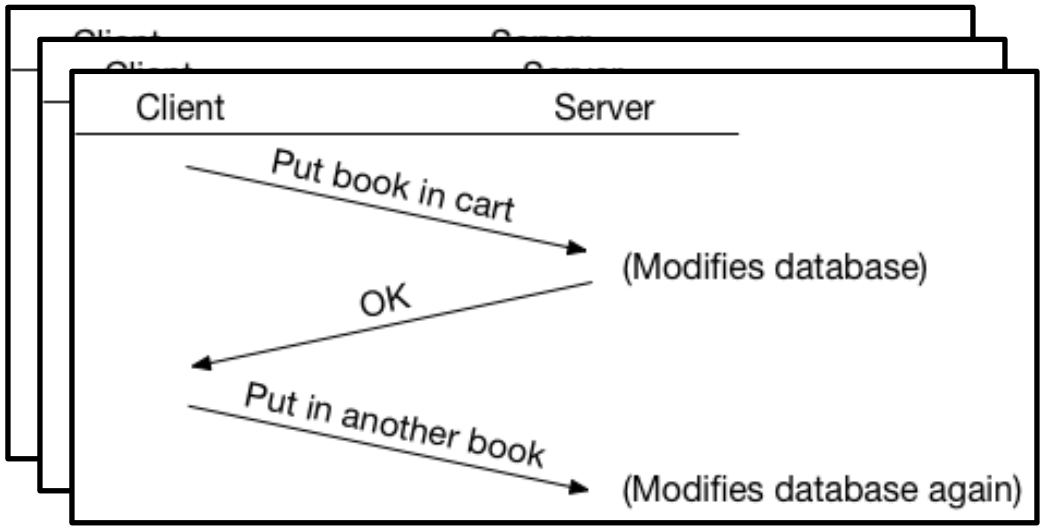

Google
Gmail
More
COMPOSE
Inbox (7)
Starred
Drafts
Sent Mail
Search people...
Jenny Kang
Peter H
Primary
Social 3 new
Google+, YouTube, Emi...
Promotions 2 new
Google Offers, Zagat
Updates 2 new
Shoehop, Blitz Air
Google+ new You were tagged in 3 photos on Google+ - Google+ You were tagged in three pl
YouTube new LauraBlack just uploaded a video. - Jess, have you seen the video LauraBlack u
Emily Million (Google+) new [Knitting Club] Are we knitting tonight? - [Knitting Club] Are we knitting tonight?
Sean Smith (Google+) Photos of the new pup - Sean Smith shared an album with you. View album be tho
Google+ Kate Baynham shared a post with you - Follow and share with Kate by adding her
Google+ Danielle Hoodhood added you on Google+ - Follow and share with Danielle by
YouTube Just for You From YouTube: Daily Update - Jun 19, 2013 - Check out the latest

Session implementation

- How to link HTTP requests to a session?

Turn this … into this



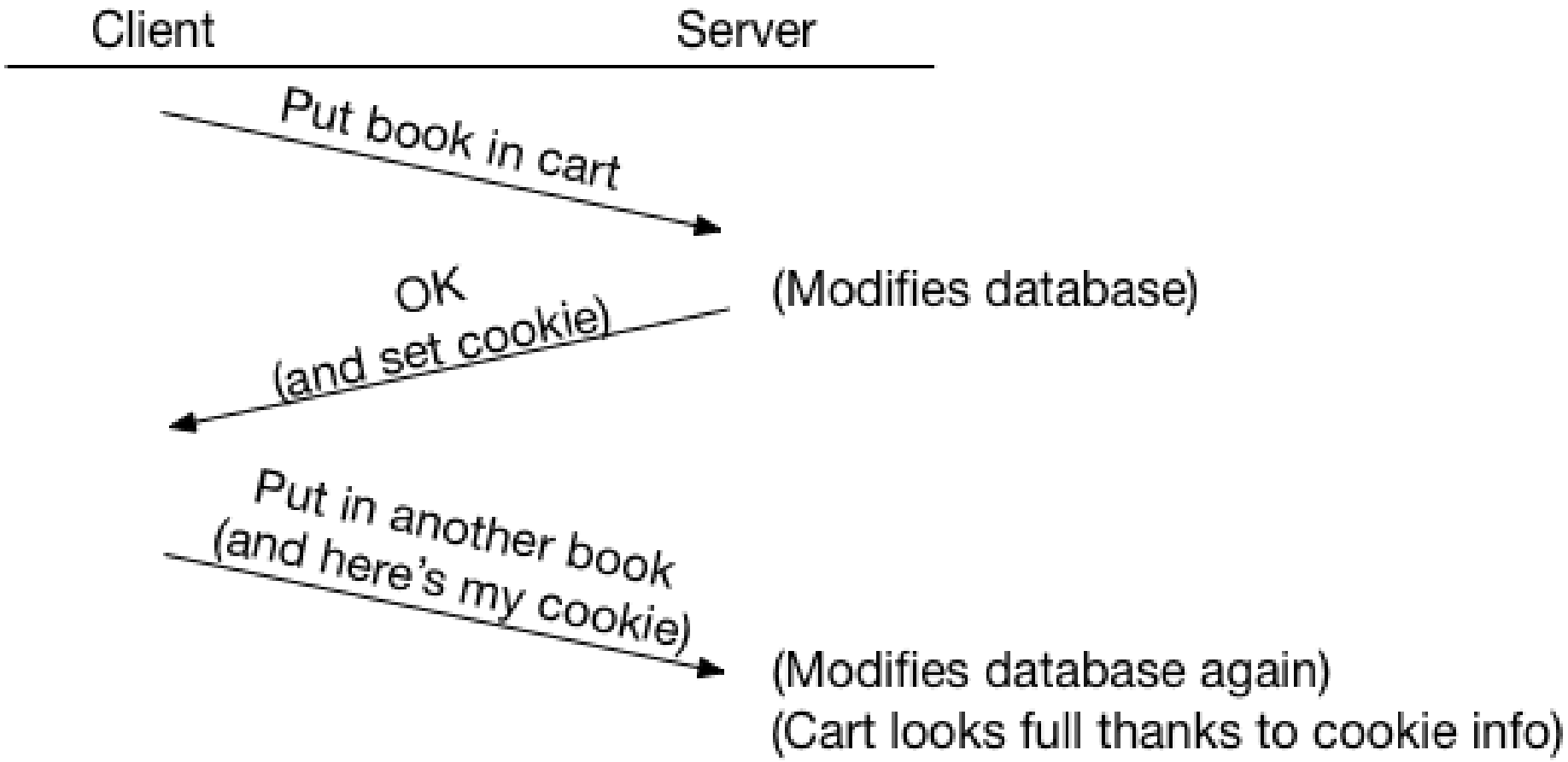


Cookies

- Cookies are small files on client machine
  - Carry state between HTTP requests
  - Contain key/value pairs

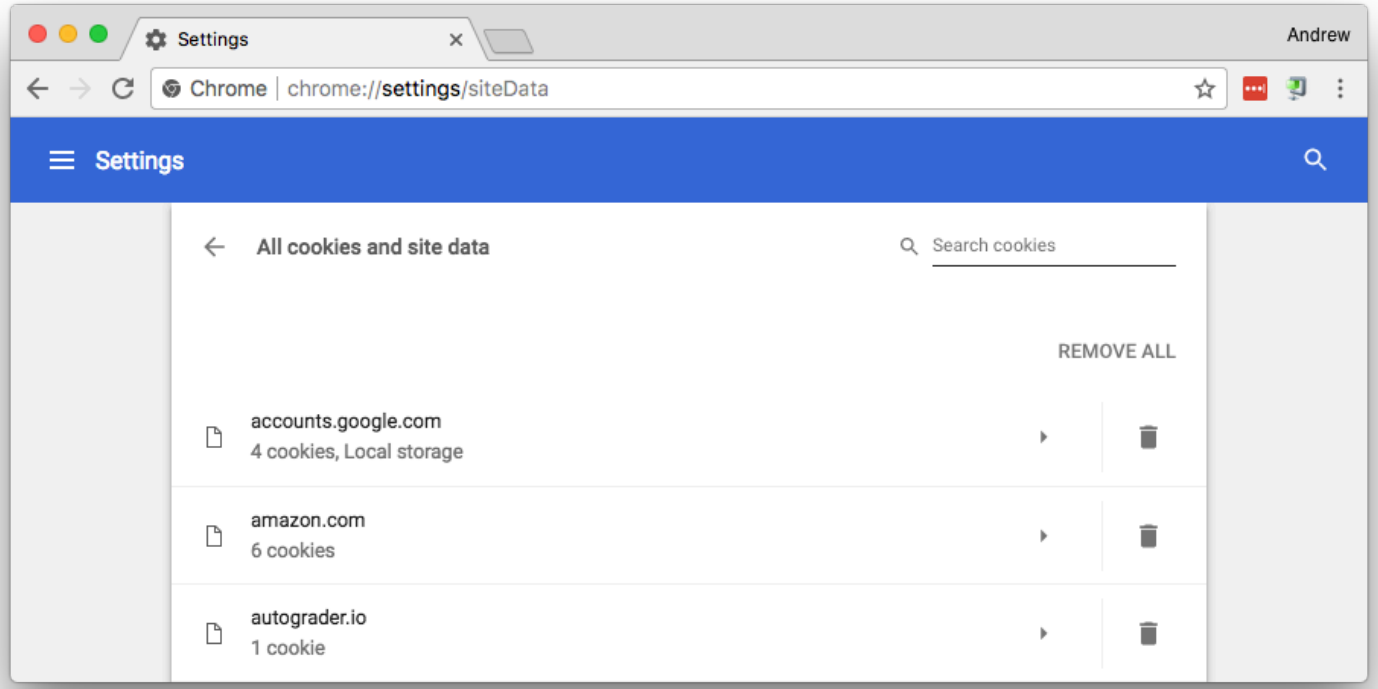
- According to the lore, the name originates from the story of Hansel and Gretel, who were able to mark their trail through a dark forest by dropping cookie crumbs behind them.

Cookies


Client
Server
Put book in cart
OK
(and set cookie)
(Modifies database)
Put in another book
(and here's my cookie)
(Modifies database again)
(Cart looks full thanks to cookie info)

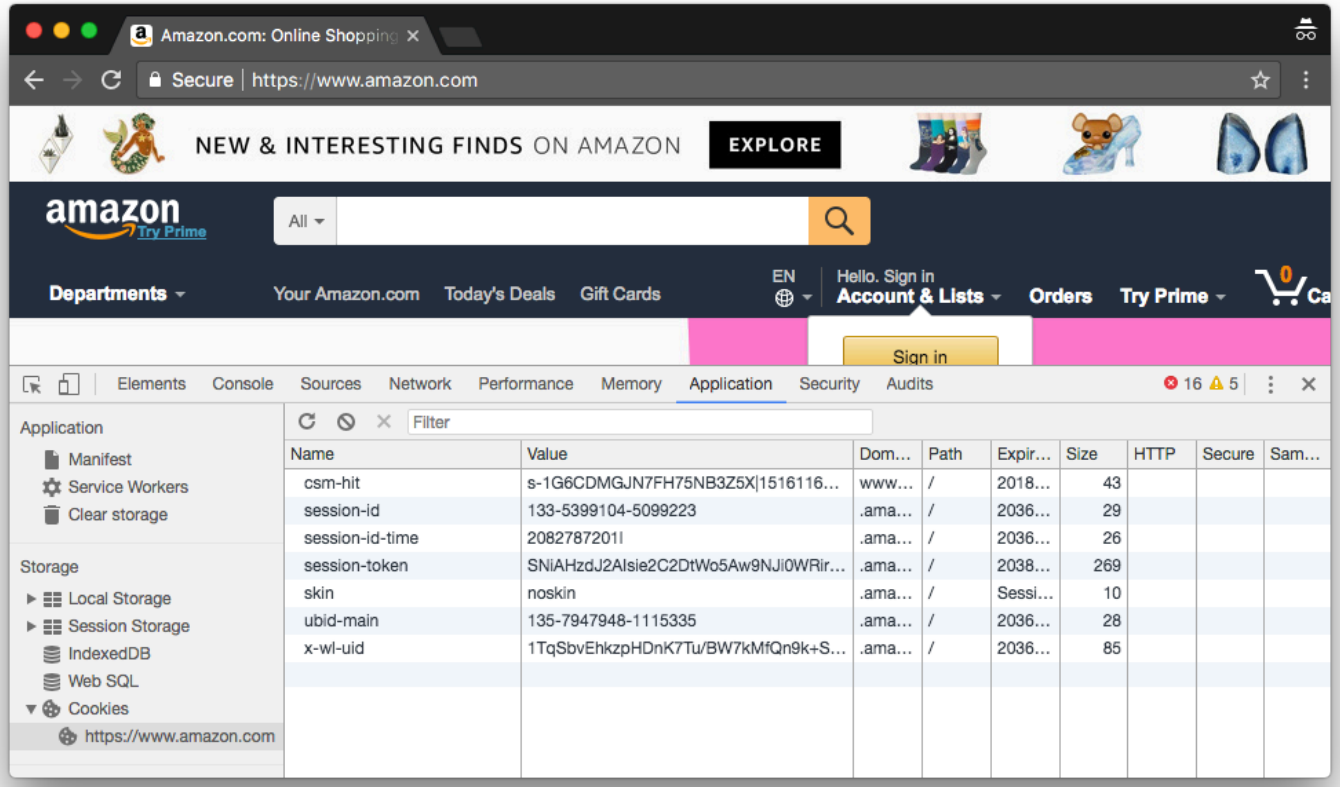
Example: cookies in Chrome

- See all cookies by browsing to `chrome://settings/siteData`
- Take a look at the cookies on your own laptop



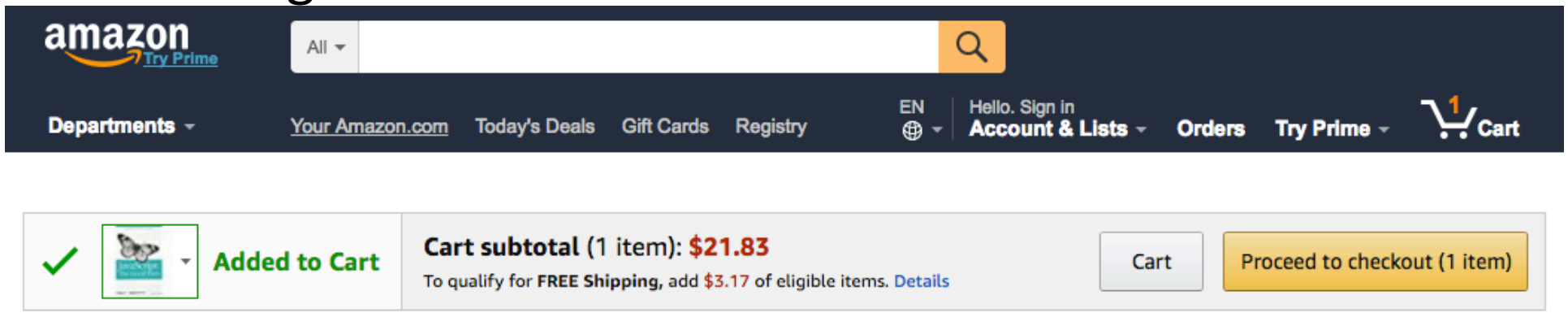
Example: cookies in Chrome

- Browse to https://www.amazon.com/
- Settings / Advanced / Content settings / Cookies / See all cookies



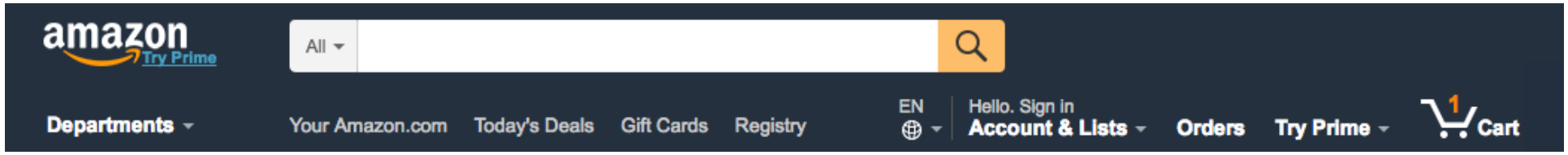
Example: shopping cart

- Browse to https://www.amazon.com/
- Add something to the cart



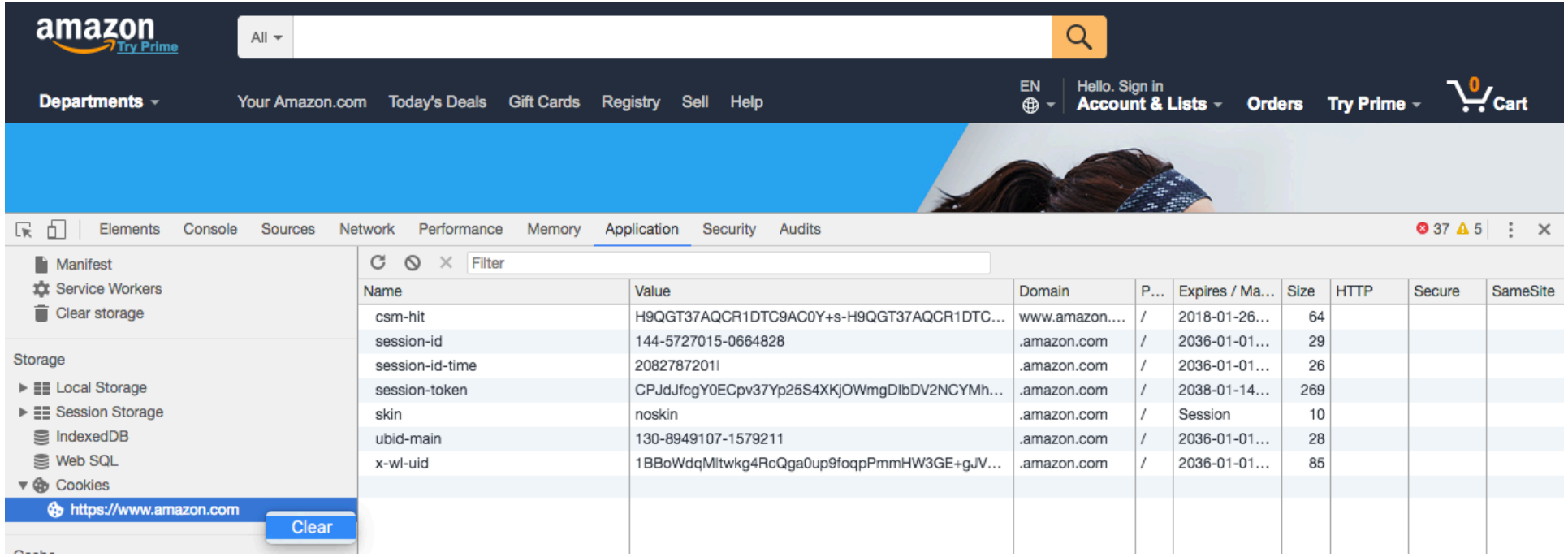
Example: shopping cart

- Again, browse to https://www.amazon.com/
- Cart has one item, even though we're on a different page



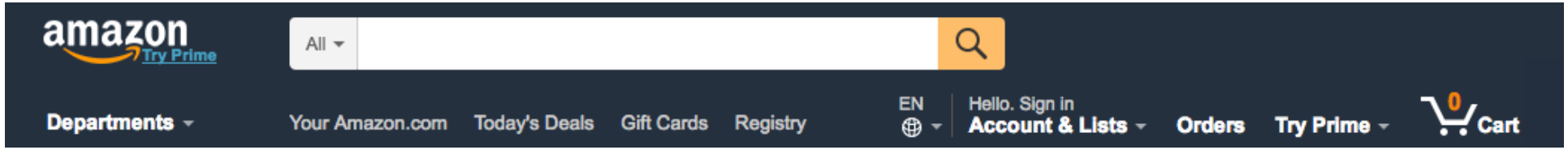
Example: shopping cart

- Clear cookies using the developer console



Example: shopping cart

- Again, browse to https://www.amazon.com/
- Cart appears empty



Cookie content

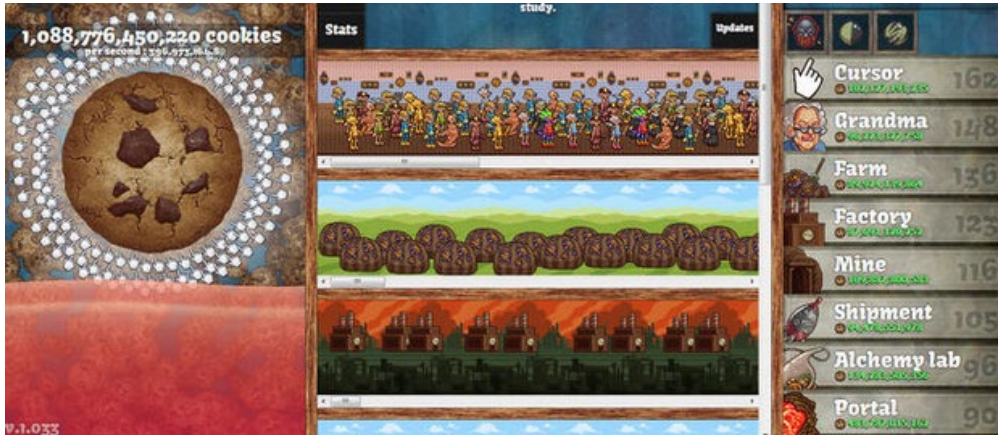
- **Name** is up to the server
- **Value** is up to server: encrypted? OK!
- **Domain** used by browser per-domain, total limits
- **Path** specifies scope of cookie
  - / vs. /cart/ or whatever
- **Expiration** tells client when to delete
- **Secure** is how cookie may be transmitted

Name	Value	Domain	Path	Expires / Max…	Size	HTTP	Secure	SameSite
csm-hit	6SMS3JE6TN8HC9JXT1BM+s-D02JSKPWCWCEQ1JBFH…	www.amazon.com	/	2018-01-23T…	64			
session-id	133-7499895-4230864	.amazon.com	/	2036-01-01T…	29			
session-id-time	2082787201l	.amazon.com	/	2036-01-01T…	26			
session-token	Z7BBJabdxLhufMLF1UNiZK75ibXyTodM3OuusYgiO7xCgR…	.amazon.com	/	2038-01-11T…	269			
skin	noskin	.amazon.com	/	Session	10			
spblockdate	1516117231080	www.amazon.com	/	2028-01-14T…	24			
ubid-main	132-0740974-0237934	.amazon.com	/	2036-01-01T…	28			
x-wl-uid	1gaw0tztjcgWgcjkDQSl9Hr4AkyB33hIw9lk44qFtLMtimrpn…	.amazon.com	/	2036-01-01T…	85			

Cookie content

- Cookies only over-writable by same domain and path
- Enforced by browser

- Store a username, session ID, etc. in the cookie
- Use session ID or username to do a database lookup
  - Shopping card content, news feed items, etc.



Cookie transfer: Server -> Client

- Headers include request to `Set-Cookie`
- These are exactly (some of) the cookies we saw in Chrome

```
$ curl --verbose --user-agent "Mozilla/5.0" https://www.amazon.com/ >
/dev/null
...
< Set-Cookie: skin=noskin; path=/; domain=.amazon.com
< Set-Cookie: session-id=130-5594428-2333702; Domain=.amazon.com;
Expires=Tue, 01-Jan-2036 08:00:01 GMT; Path=/
< Set-Cookie: session-id-time=2082787201l; Domain=.amazon.com; Expires=Tue,
01-Jan-2036 08:00:01 GMT; Path=/
```

Name	Value	Domain	Path	Expires / Max...	Size	HTTP	Secure	SameSite
csm-hit	6SMS3JE6TN8HC9JXT1BM+s-D02JSKPWCWCEQ1JBFH...	www.amazon.com	/	2018-01-23T...	64			
session-id	133-7499895-4230864	.amazon.com	/	2036-01-01T...	29			
session-id-time	2082787201l	.amazon.com	/	2036-01-01T...	26			
session-token	Z7BBJabdxLhufMLF1UNiZK75ibXyTodM3OuusYgiO7xCgR...	.amazon.com	/	2038-01-11T...	269			
skin	noskin	.amazon.com	/	Session	10			
spblockdate	1516117231080	www.amazon.com	/	2028-01-14T...	24			
ubid-main	132-0740974-0237934	.amazon.com	/	2036-01-01T...	28			
x-wl-uid	1gaw0tztjcgWgcjkDQSl9Hr4AkyB33hIw9lk44qFtLMtimrpn...	.amazon.com	/	2036-01-01T...	85			

Cookie transfer: Client -> Server

```
$ curl --verbose --user-agent "Mozilla/5.0"
https://www.amazon.com/ ... > /dev/null
...
> HTTP 1.1 GET /
> Cookie: skin=noskin; session-id=130-5594428-2333702;
session-id-time=2082787201l;
```

Cookie transfer

- Cookies add to HTTP overhead
- Again, best practice is to store a small amount of data in the cookie
- Use that data to do a database lookup on the server side

Saving cookies

- Your browser saves cookies by default
- Tell `curl` to save cookies with `--cookie-jar`

```
$ curl --verbose --user-agent "Mozilla/5.0"
  --cookie-jar cookies.txt https://www.amazon.com/
  > /dev/null

$ cat cookies.txt
.amazon.com TRUE / FALSE 0 skin noskin
.amazon.com TRUE / FALSE 2082787201 session-id 147-4402398-5757441
.amazon.com TRUE / FALSE 2082787201 session-id-time 20827872011
```

Protecting cookies

- Anyone that has your cookies can convince server it's you

- **Session Hijacking** is when an adversary steals your session
  - If they can copy cookies from your computer, they can take your identify for those websites for which you have cookies downloaded
  - Consider: when you login, you get a cookie that the server uses to vet you as “already-logged in.”

- Use HTTPS so that Cookie headers aren’t sent in plaintext!

Encrypted cookie transfer

- If I have your cookies, I can steal your session
  - To the server, I look just like you!
- Prevent this with cookies that can only be transmitted over HTTPS

Name	Value	Domain	P…	Expires / Max…	Size	HTTP	Secure
a-ogbcbff	1	.amazon.com	/	2018-01-16T…	10		
at-main	Atza|IwEBIL-ZdP_TnX55gVrEdKvscQRM4Tjoq0cTj_bivnQG…	.amazon.com	/	2038-01-11T…	424	✓	✓
aws-ubid-main	102-1422888-1085586	.amazon.com	/	2086-02-03T…	32	✓	✓
csm-hit	s-MZBVZTC3VF120DMVJDF6|1516118415619	www.amazon.com	/	2018-01-23T…	43		
lc-main	en_US	.amazon.com	/	2038-01-11T…	12		
sess-at-main	"bWhsxGV15YSI5RDUGleDQGcU9k+PLNN0MI3JPNsPj28="	.amazon.com	/	Session	58	✓	✓
session-id	142-6976303-8296237	.amazon.com	/	2036-01-01T…	29		
session-id-time	2082787201l	.amazon.com	/	2036-01-01T…	26		
session-token	OCpqc+N1p2eiUvVrZI+3wUqsRi6Fm3I6JnSMhC19oN5kve…	.amazon.com	/	2038-01-11T…	281		
skin	noskin	.amazon.com	/	Session	10		
spblockdate	1514917833143	www.amazon.com	/	2027-12-31T…	24		
sst-main	Sst1|PQGjiLW9hWUjZlPblHPlPXH8CAeBAYdLmz4cPvGgEb…	.amazon.com	/	2038-01-11T…	295	✓	✓
ubid-main	132-7633599-2974057	.amazon.com	/	2036-01-01T…	28		
x-main	"JqL@@cu6yh0xuDwQrAfLeP9PI@9TlcZgRYXh08zDQC7gV…	.amazon.com	/	2038-01-11T…	72		
x-wl-uid	1qpmQHYEzowtwk8DHEWu0Bfr4FGX+O579dYLEygZC8kyj…	.amazon.com	/	2036-01-01T…	149		

Thought Question

- When you log into johnmail.com, it sets a cookie "userid=200".
- What might be insecure about this?
- How could you fix it?

Encrypted cookie values

- If the client has a copy of cookies set by the server, then the client can manipulate the server
- Prevent this with encrypted cookie content
  - Only the server can decrypt
- **Note:** This is *separate* from using HTTPS to *transmit* the cookie!

Name	Value	Domain	P...	Expires / Max...	Size	HTTP	Secure
a-ogbcbff	1	.amazon.com	/	2018-01-16T...	10		
at-main	Atza|IwEBIL-ZdP_TnX55gVrEdKvscQRM4Tjoq0cTj_bivnQG...	.amazon.com	/	2038-01-11T...	424	✓	✓
aws-ubid-main	102-1422888-1085586	.amazon.com	/	2086-02-03T...	32	✓	✓
csm-hit	s-MZBVZTC3VF120DMVJDF6|1516118415619	www.amazon.com	/	2018-01-23T...	43		
lc-main	en_US	.amazon.com	/	2038-01-11T...	12		
sess-at-main	"bWhsxGV15YSI5RDUGIeDQGcU9k+PLNN0MI3JPNsPj28="	.amazon.com	/	Session	58	✓	✓
session-id	142-6976303-8296237	.amazon.com	/	2036-01-01T...	29		
session-id-time	2082787201l	.amazon.com	/	2036-01-01T...	26		
session-token	OCpqc+N1p2eiUvVrZI+3wUqsRi6Fm3I6JnSMhC19oN5kve...	.amazon.com	/	2038-01-11T...	281		
skin	noskin	.amazon.com	/	Session	10		
spblockdate	1514917833143	www.amazon.com	/	2027-12-31T...	24		
sst-main	Sst1|PQGjiLW9hWUjZIPbIHPIPXH8CAeBAYdLmz4cPvGgEb...	.amazon.com	/	2038-01-11T...	295	✓	✓
ubid-main	132-7633599-2974057	.amazon.com	/	2036-01-01T...	28		
x-main	"JqL@@cu6yh0xuDwQrAfLeP9PI@9TlcZqRYXh08zDQC7qV...	.amazon.com	/	2038-01-11T...	72		

Session vs. permanent cookies

- Session cookies are deleted by the client shuts down
  - Close the tab or quit the browser
- Permanent cookies have explicit expiration dates

Name	Value	Domain	P…	Expires / Max…	Size	HTTP	Secure
a-ogbcbff	1	.amazon.com	/	2018-01-16T…	10		
at-main	Atza|IwEBIL-ZdP_TnX55gVrEdKvscQRM4Tjoq0cTj_bivnQG…	.amazon.com	/	2038-01-11T…	424	✓	✓
aws-ubid-main	102-1422888-1085586	.amazon.com	/	2086-02-03T…	32	✓	✓
csm-hit	s-MZBVZTC3VF120DMVJDF6|1516118415619	www.amazon.com	/	2018-01-23T…	43		
lc-main	en_US	.amazon.com	/	2038-01-11T…	12		
sess-at-main	"bWhsxGV15YSI5RDUGleDQGcU9k+PLNN0MI3JPNsPj28="	.amazon.com	/	Session	58	✓	✓
session-id	142-6976303-8296237	.amazon.com	/	2036-01-01T…	29		
session-id-time	2082787201l	.amazon.com	/	2036-01-01T…	26		
session-token	OCpqc+N1p2eiUvVrZI+3wUqsRi6Fm3I6JnSMhC19oN5kve…	.amazon.com	/	2038-01-11T…	281		
skin	noskin	.amazon.com	/	Session	10		
spblockdate	1514917833143	www.amazon.com	/	2027-12-31T…	24		
sst-main	Sst1|PQGjiLW9hWUjZlPblHPlPXH8CAeBAYdLmz4cPvGgEb…	.amazon.com	/	2038-01-11T…	295	✓	✓
ubid-main	132-7633599-2974057	.amazon.com	/	2036-01-01T…	28		
x-main	"JqL@@cu6yh0xuDwQrAfLeP9PI@9TlcZgRYXh08zDQC7gV…	.amazon.com	/	2038-01-11T…	72		
x-wl-uid	1qpmQHYEzowtwk8DHEWu0Bfr4FGX+O579dYLEygZC8kyj…	.amazon.com	/	2036-01-01T…	149		

Third-party cookies

- Page may contain objects from many sources
  - Scripts, images, etc.
- These 3rd-party objects set and get cookies
- Example: nytimes.com

```
<html>
  <head>
    <script
      src="https://tags.bluekai.com/site/50550?ret=js&limit=1"
      type="text/javascript"
    >
    </script>
  </head>
  <body>
    <img src="https://static01.nyt.com/newsgraphics/2018/01/18/shutdown-deal-factions/assets/images/lindsey-graham.png">
  </body>
</html>
```

Third-party cookies

- Cookies have a domain
- *First-party cookie*: domain is the same as the domain of the page you are on
- *Third-party cookie*: domain is different
- Example from nytimes.com

Name	Value	Domain	P…	Expires / Ma…	Size	HTTP	Secure	SameSite
IDE	AHWqTUle_KCiqXNrNt3LWAFbPHlkNNCKdka6oVRxF…	.doubleclick.net	/	2020-01-16…	67	✓		
UID	15718486a24320a1606e8cg1516118895	.scorecardres…	/	2020-01-06…	36			
UIDR	1516118895	.scorecardres…	/	2020-01-06…	14			
__gads	ID=a2a38e066efb072b:T=1516118895:S=ALNI_MYy9…	.nytimes.com	/	2020-01-16…	75			
__utma	69104142.228179094.1516118895.1516118895.15161…	.nytimes.com	/	2020-01-16…	59			
__utmb	69104142.1.10.1516118896	.nytimes.com	/	2018-01-16…	30			
__utmc	69104142	.nytimes.com	/	Session	14			
__utmt	1	.nytimes.com	/	2018-01-16…	7			
__utmz	69104142.1516118896.1.1.utmcsr=(direct)|utmccn=(di…	.nytimes.com	/	2018-07-18…	75			

Example: Google third party cookies

- One of the main advertising cookies on non-Google sites is named `IDE` and is stored in browsers under the domain `doubleclick.net`.
  - https://www.google.com/policies/technologies/types/
- Visit nytimes.com and view cookies:

Name	Value	Domain	P…	Expires / Ma…	Size	HTTP	Secure	SameSite
IDE	AHWqTUle_KCiqXNrNt3LWAFbPHlkNNCKdka6oVRxF…	.doubleclick.net	/	2020-01-16…	67	✓		
UID	15718486a24320a1606e8cg1516118895	.scorecardres…	/	2020-01-06…	36			
UIDR	1516118895	.scorecardres…	/	2020-01-06…	14			
__gads	ID=a2a38e066efb072b:T=1516118895:S=ALNI_MYy9…	.nytimes.com	/	2020-01-16…	75			
__utma	69104142.228179094.1516118895.1516118895.15161…	.nytimes.com	/	2020-01-16…	59			
__utmb	69104142.1.10.1516118896	.nytimes.com	/	2018-01-16…	30			
__utmc	69104142	.nytimes.com	/	Session	14			
__utmt	1	.nytimes.com	/	2018-01-16…	7			
__utmz	69104142.1516118896.1.1.utmcsr=(direct)|utmccn=(di…	.nytimes.com	/	2018-07-18…	75			

Example: Google third party cookies

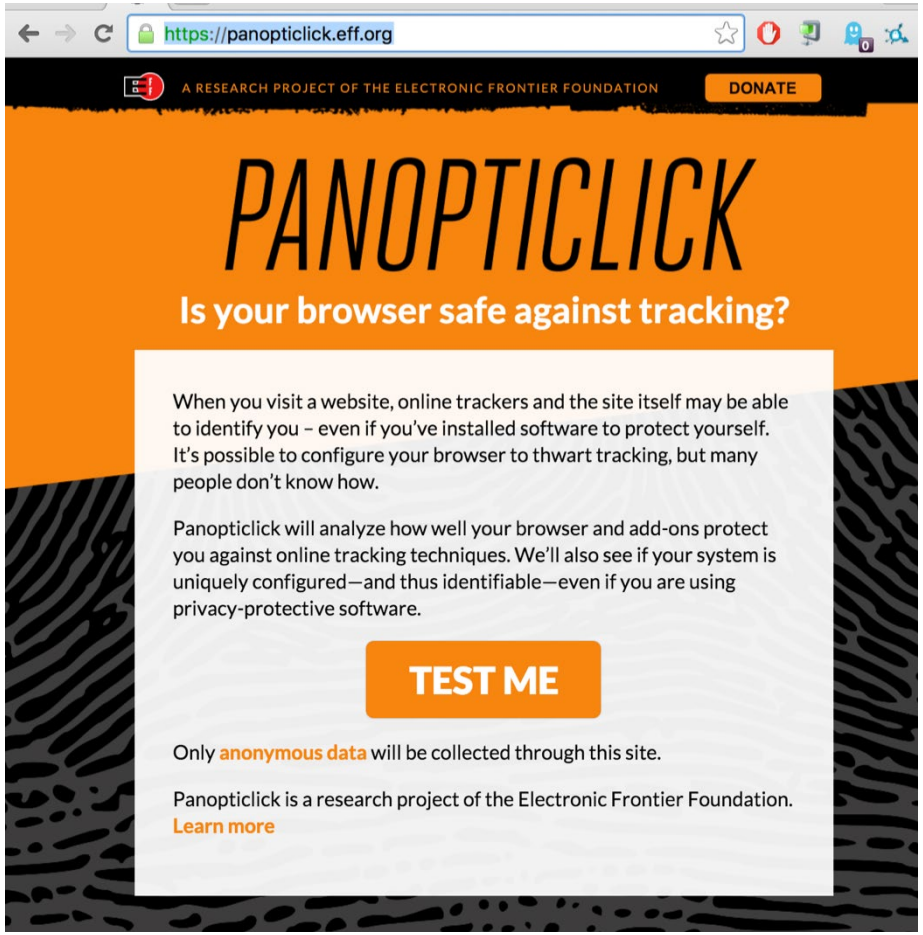
- You type nytimes.com into your browser
- Browser issues `GET` request to nytimes.com
  - Includes nytimes.com cookies
- Browser receives HTML for nytimes.com
  - HTML includes some JavaScript via the `<script>` tag
- Browser executes JavaScript included by nytimes.com
  - JS code figures out you are on nytimes.com, e.g., with `window.location.href`
  - JS codes initiates a request to doubleclick.net
- Browser issues `GET` request to doubleclick.net
  - Includes doubleclick.net cookie
  - Appends your current location (nytimes.com) to the URL
- Now, doubleclick.net (AKA Google) knows you visited nytimes.com

Who uses third-party cookies?

- Companies that sell ads directly
  - Google and Facebook
- Companies that sell information about you
  - Acxiom "one of the biggest companies you've never heard of" ($1B+)
- “If you're not paying for the product, then you *are* the product”
  - That's mostly how the business side of the web is structured

Checking what you send to trackers

- https://panopticlick.eff.org/

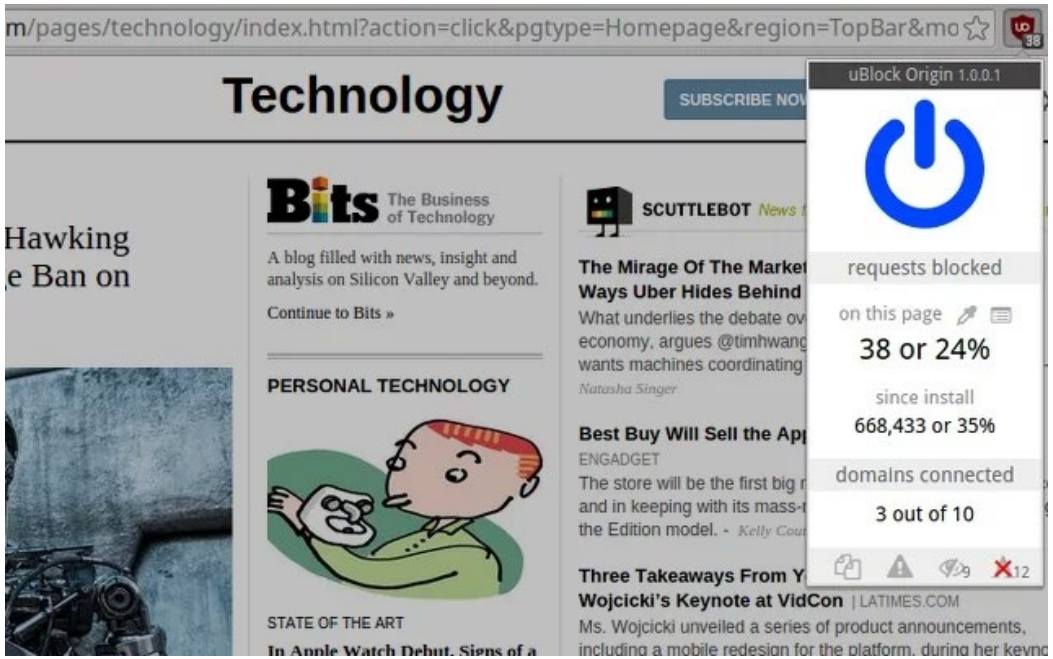


Browser fingerprinting

- Browser fingerprinting attempts to uniquely identify your browser using information other than cookies
  - User Agent
  - Time zone
  - Fonts
  - Language
  - And lots more
- Anti-fingerprinting options announced by Firefox, Chrome, Safari
  - As of fall 2019

Avoiding trackers

- Add-ons like uBlock Origin, Privacy Badger, Brave, Disconnect or ScriptNo block trackers
- Monitors embedded links and blacklists trackers



Discussion

- Many web companies make their money from information about users
- In exchange, they give you a "free" service (email, web search, a platform for gossip, etc.)
- Is it OK to block trackers?

Cookie example

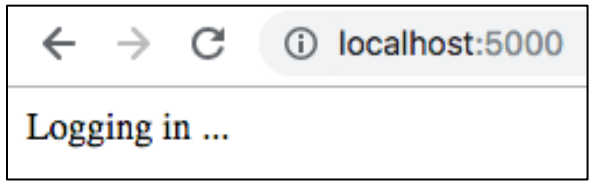
```
import flask
app = flask.Flask(__name__)

app.secret_key = b'uAy\x9d\x08[\x12\x8d\x9d\x1f\xbar\x86A\x9fpQy4\x05)v04'

@app.route('/')
def index():
    if "user" in flask.session:
        user = flask.session["user"]
        app.logger.debug("Get user=%s", user)
        return "<html><body>Hello {}</body></html>".format(user)
    else:
        flask.session["user"] = "awdeorio"
        app.logger.debug("Set user=%s", user)
        return "<html><body>Logging in ...</body></html>"

if __name__ == '__main__':
    app.run(debug=True)
```

Cookie example

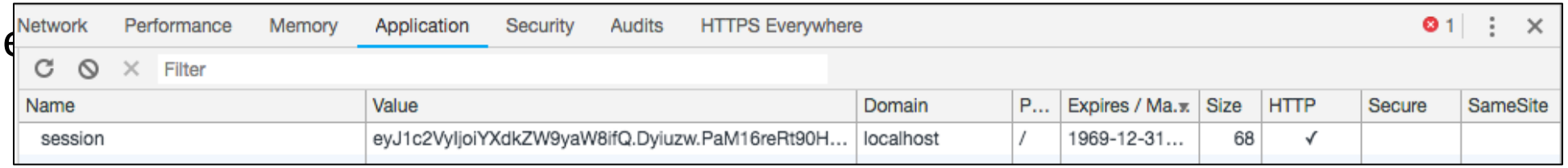
- Start server

```
$ python3 test.py
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
[2019-01-22 08:40:31,790] DEBUG in test: Set user=awdeorio
127.0.0.1 - - [22/Jan/2019 08:40:31] "GET / HTTP/1.1" 200 -
```

- Browse to http://localhost:5000/



- See



Cookie encryption key

```
import flask
app = flask.Flask(__name__)

app.secret_key = b'uAy\x9d\x08[\x12\x8d\x9d\x1f\xbar\x86A\x9fpQy4\x05)v04'

# ...
```

- Session cookies are encrypted
- The server has the encryption key

Cookie encryption key

- How to generate a secure encryption key?

- Need a cryptographically secure random number.
  - If a "random" number is in any way predictable, then a hacker could guess it and masquerade as any user!

- Generate a Python string

- `$ python3 -c "import os; print(os.urandom(24))"`
  `b'uAy\x9d\x08[\x12\x8d\x9d\x1f\xbar\x86A\x9fpQy4\x05)v04'`

Further reading

- Good reference on cookies
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies