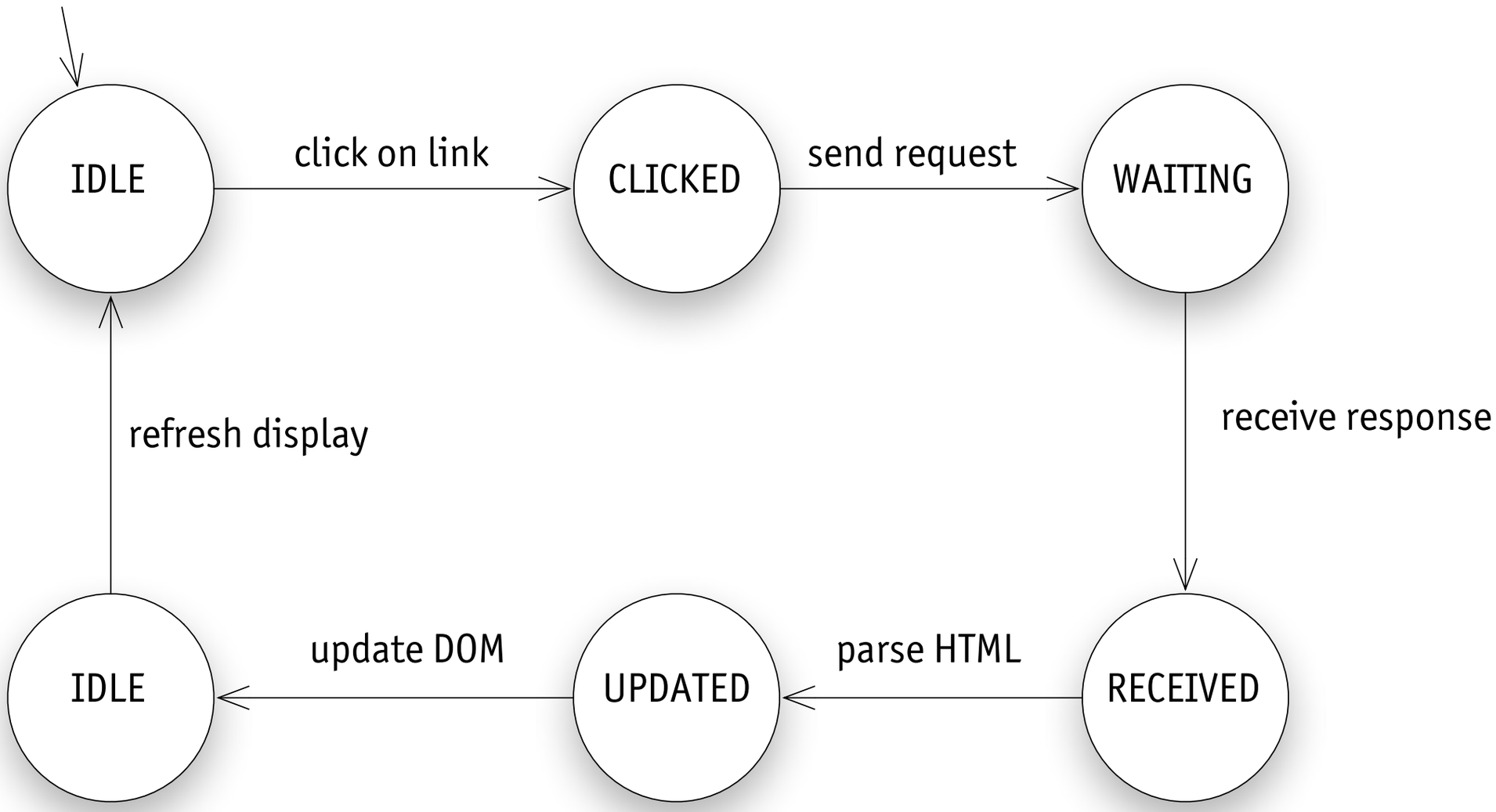
software studio

http:
hypertext transfer protocol

Daniel Jackson

what browser does



- basic page load is synchronous
- when page arrives, often loads others (images, css, js)
- complications: caching, asynchronous calls, events

http protocol

simplified protocol

session ::= request response

request ::= requestLine header+ [body]

requestLine ::= method path

response ::= status header+ [body]

what headers contain

› info about request or response, or about body

what body contains

› in request: form data, uploaded files

› in response: resource

sample http interactions

Headers Content

Request URL: http://www.google.com/search?client=safari&rls=en&q=photography&ie=UTF-8&oe=UTF-8
Request Method: GET
Status Code: 200 OK

▼ Request Headers view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50

▼ Query String Parameters view URL encoded

client: safari
rls: en
q: photography
ie: UTF-8
oe: UTF-8

▼ Response Headers view source

Cache-Control: private, max-age=0
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Mon, 17 Oct 2011 01:13:54 GMT
Expires: -1
Server: gws
Transfer-Encoding: Identity
X-Xss-Protection: 1; mode=block

› GET, data passed as query string

sample http interactions

Headers | Content

Request URL: http://travel.travelocity.com/flights/InitialSearch.do
Request Method: POST
Status Code: 200 OK

▼ Request Headers view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Content-Type: application/x-www-form-urlencoded
Origin: http://www.travelocity.com
Referer: http://www.travelocity.com/472d
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50

▼ Form Data view URL encoded

Service: TRAVELOCITY
last_pgd_page: ushpnbff.pgd
entryPoint: FD
pkg_type: fh_pkg
flightType: roundtrip
subnav: form-fo
leavingFrom: boston
goingTo: london
dateTypeSelect: exactDates
leavingDate: mm/dd/yyyy
dateLeavingTime: Anytime
departDateFlexibility: 1
departure_dt: Oct
returningDate: mm/dd/yyyy
dateLeavingTime: Anytime
returnDateFlexibility: 1
arrival_dt: Feb
adults: 1
children: 0
seniors: 0
minorsAge0: ?
minorsAge1: ?
minorsAge2: ?
minorsAge3: ?
minorsAge4: ?

▼ Response Headers view source

- POST, data passed as form data

response status codes

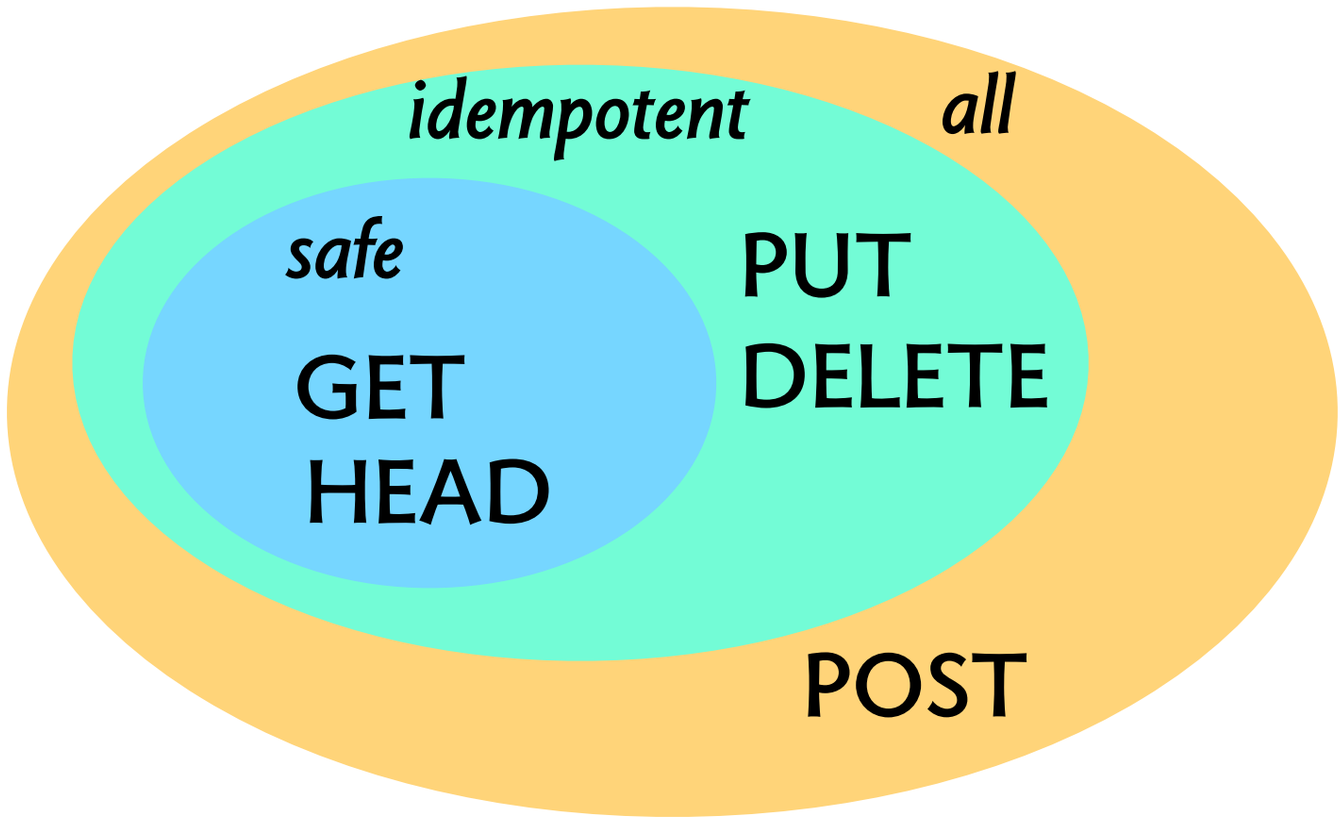
categories of codes

- 1xx informational
- 2xx success
- 3xx redirect
- 4xx client error
- 5xx server error

most common codes

- 200 OK (request succeeded, resource is in message body)
- 404 Not Found (resource doesn't exist)
- 303 See Other (resource moved, see location header)
- 500 Server Error (web app implementer messed up)

http methods



- safe: no side effects
- idempotent: doing twice same as once
- PUT vs POST: whether object is at given URI, or child of
- PUT & DELETE used in APIs but not usually in browsers

http is stateless

HTTP

› server gets request
› performs action
› generates response

previous interactions

› have no effect on server
› same request, same action

TCP: a stateful protocol

State/transition diagram of TCP protocol removed due to copyright restrictions.

Source: Stevens, W. Richard. *TCP/IP Illustrated, Vol. 1: The Protocols*. Addison-Wesley Longman, 1994, p. 576.

why stateless?

no session state

› so memory doesn't grow with number of clients
› if client dies, no problem for server

but...

› actually need session state (logged in, shopping cart)

solution

› server sends state to client as 'cookie'
› client sends it back on each request

often

› server stores session state
› sends session id in cookie

cookies in http

cookie is

› name-value pair
› expiration, path & domain

server sends

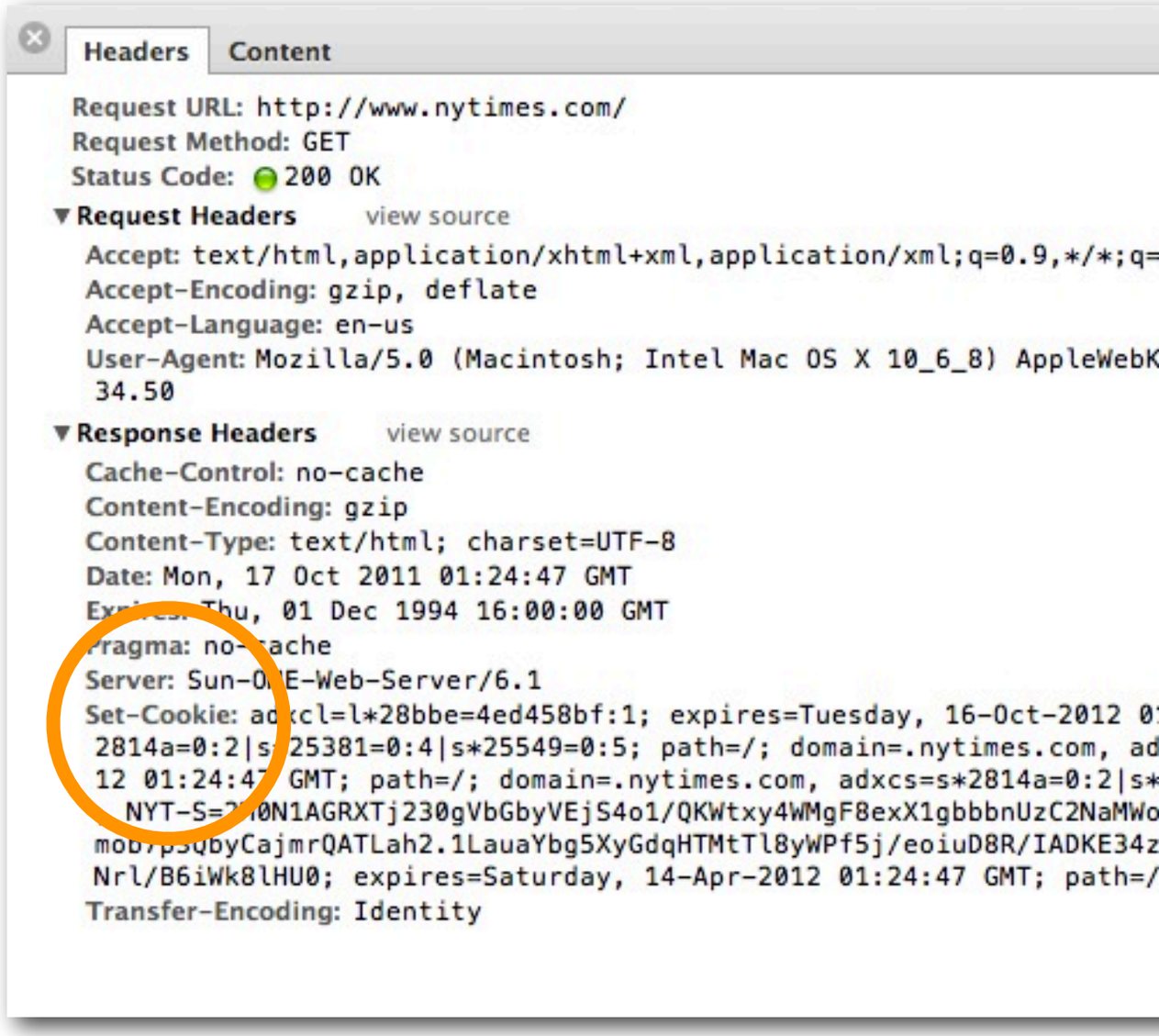
› using set-cookie header

browser sends back

› all unexpired cookies
› with matching path

expiration

› session cookies: on quit
› persistent cookies: on expire



a funny cookie tale

nytimes.com used cookies to count #articles read, so viewers just deleted cookies...

example use of cookies: logging in

step 1: user opens home page

› request includes no cookies

› response body includes no member content

step 2: user submits login form

› request is POST with user and password as data

› response includes set-cookie headers <user: dnj, login: true>

step 3: user reopens home page

› request includes all cookies for domain

› response body includes member content

how to prevent cookies being faked?

› server encrypts cookie values with secret key

6.170 Software Studio
Spring 2013