
HTTP Cookies
Photo from: http://www.flickr.com/photos/mrsmagic/1117398599

HTTP Cookies

Photo from: http://www.flickr.com/photos/mrsmagic/1117398599

Overview

1. Cookies: the simple version
2. Using cookies for managing sessions
3. Cookies: the complex version

HTTP Cookies

Photo from: http://www.flickr.com/photos/mrsmagic/1117398599/

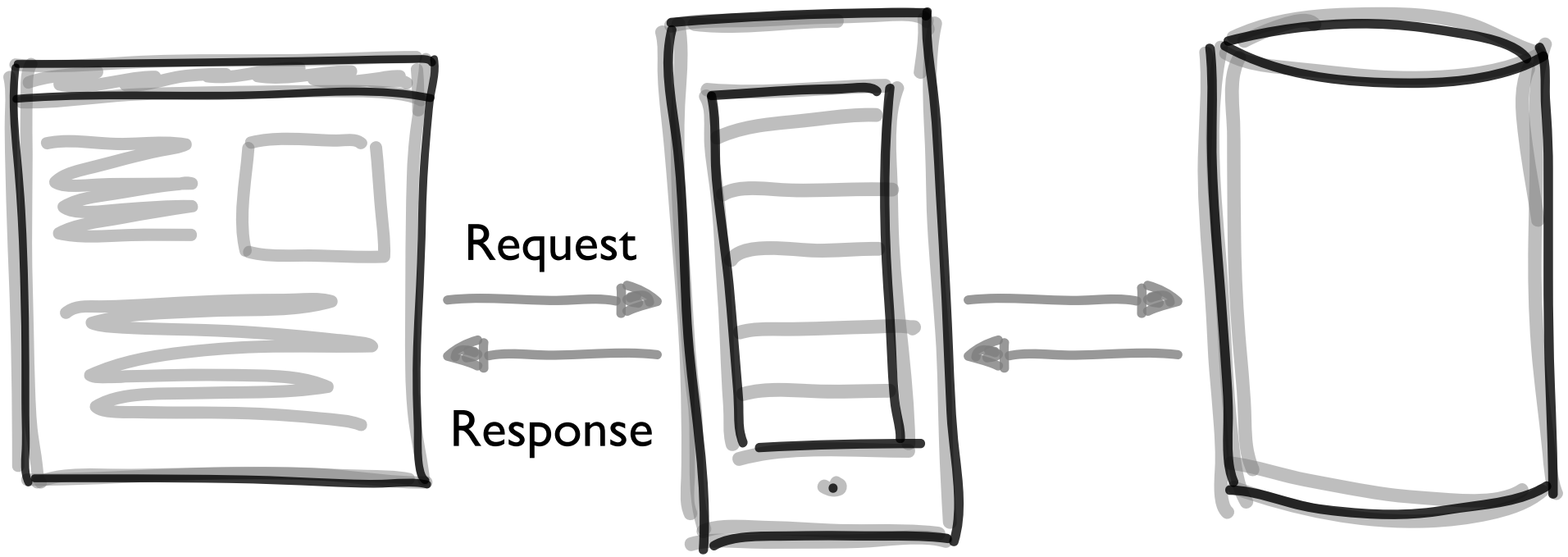
Overview

1. Cookies: the simple version
2. Using cookies for managing sessions
3. Cookies: more details

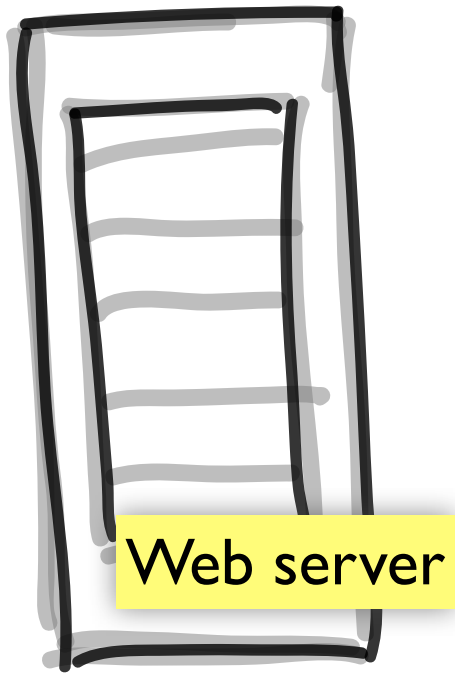
An HTTP cookie is

- Text stored on by a user agent (i.e., on the user's computer) on behalf of the server.
- This text is one or more name-value pairs containing bits of information.
- Set as part of a response from a web server (in the header of the response)
  - Set-Cookie: name=value
- Returned by the browser with each subsequent request to that server (in the header of the response)
  - Cookie: name=value

Cookies in HTTP requests and responses

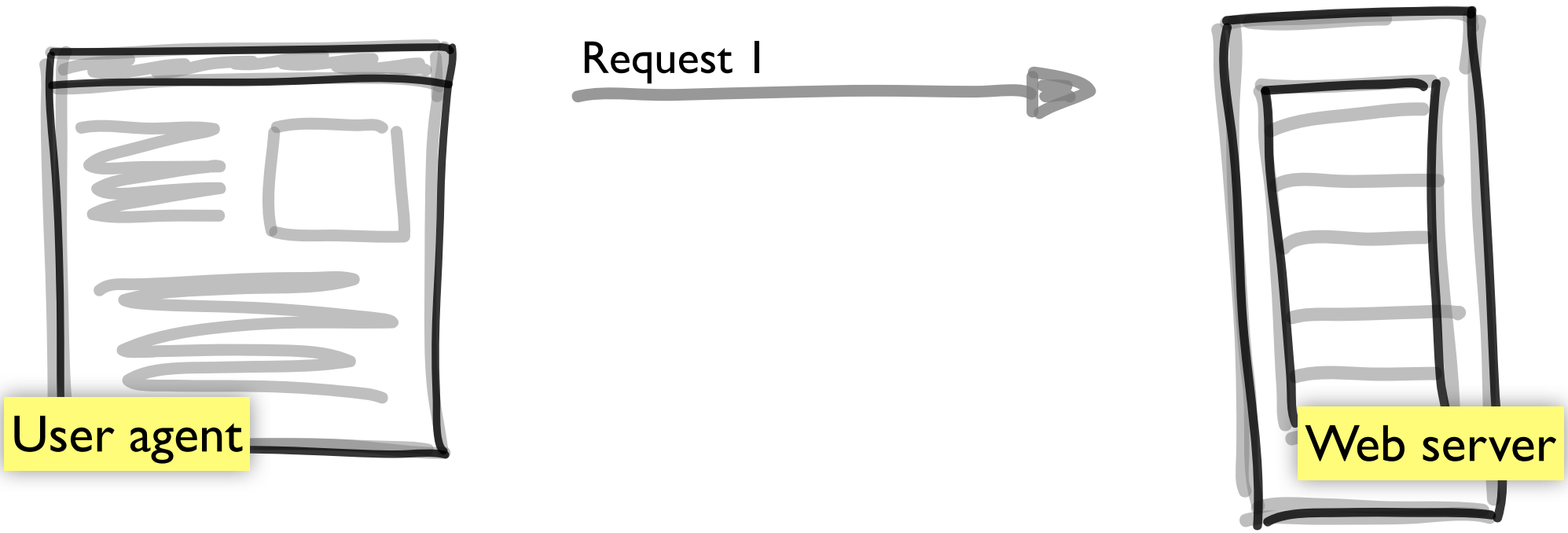


Cookies in HTTP requests and responses



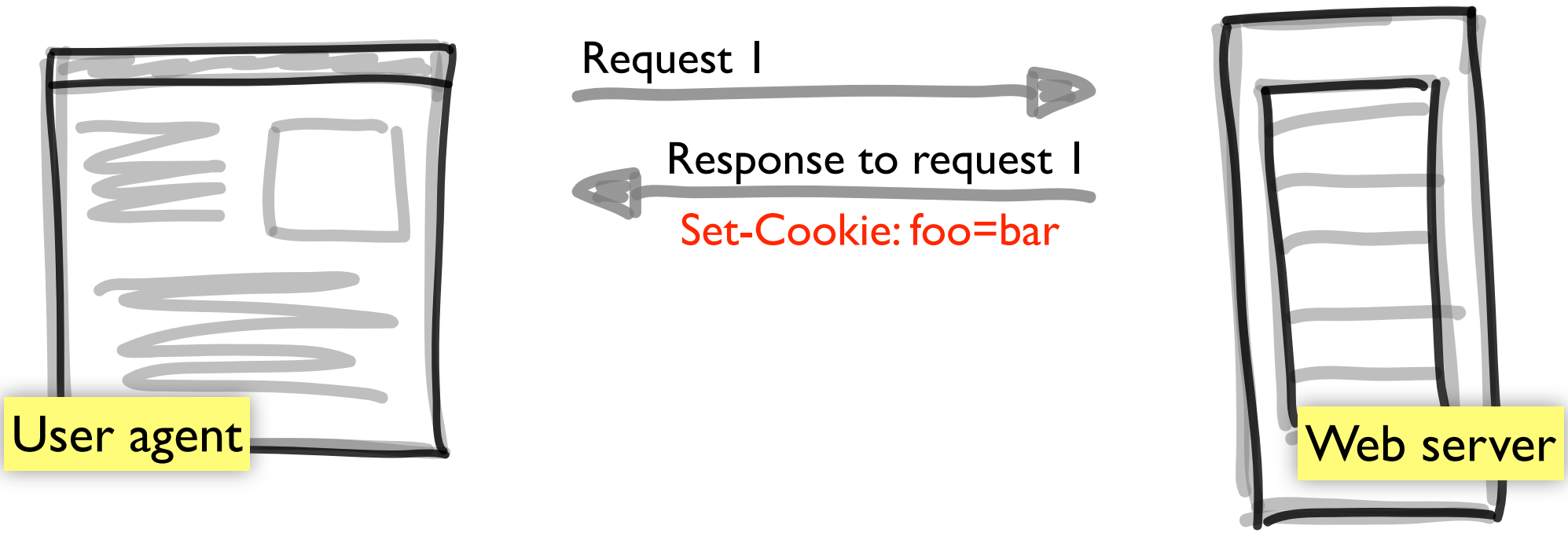
Information in HTTP headers in red

Cookies in HTTP requests and responses



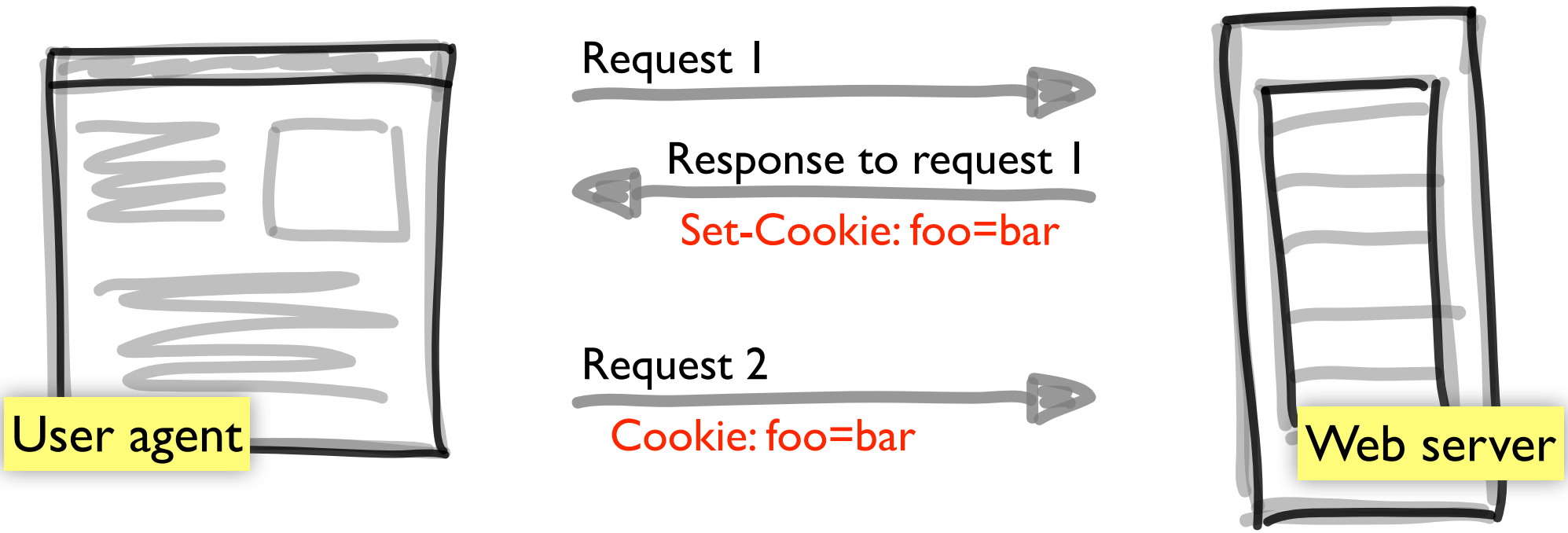
Information in HTTP headers in red

Cookies in HTTP requests and responses



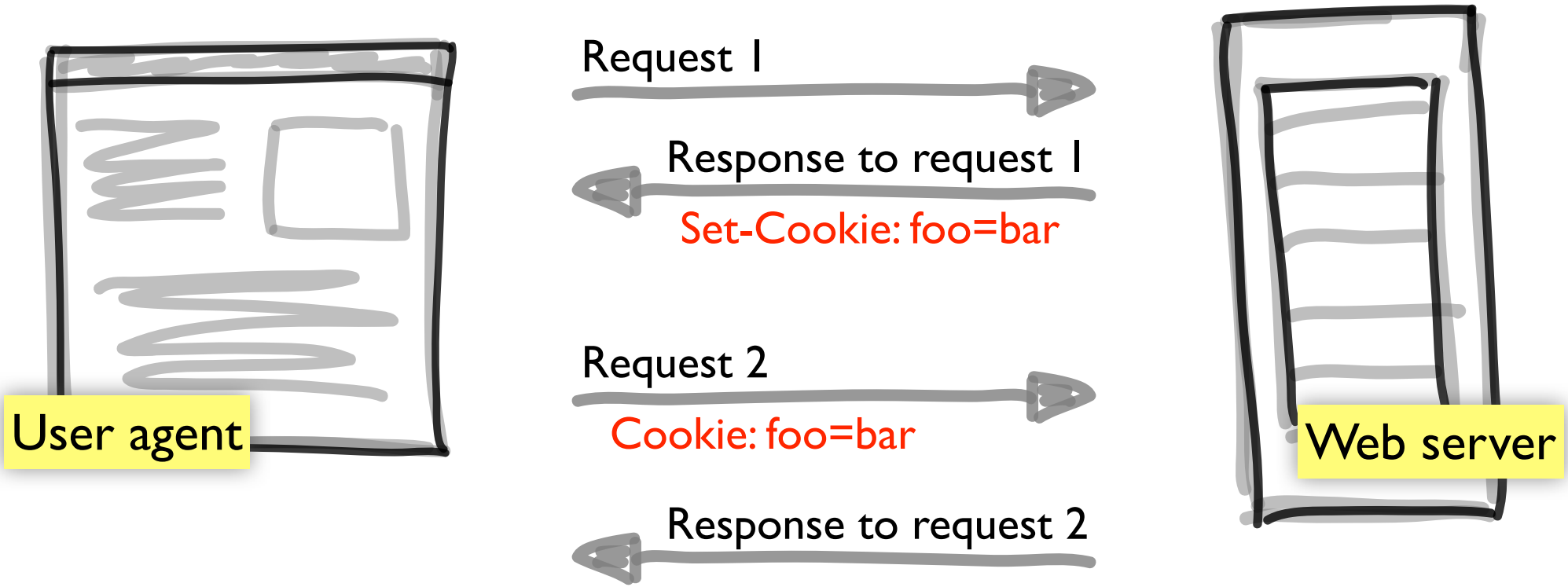
Information in HTTP headers in red

Cookies in HTTP requests and responses



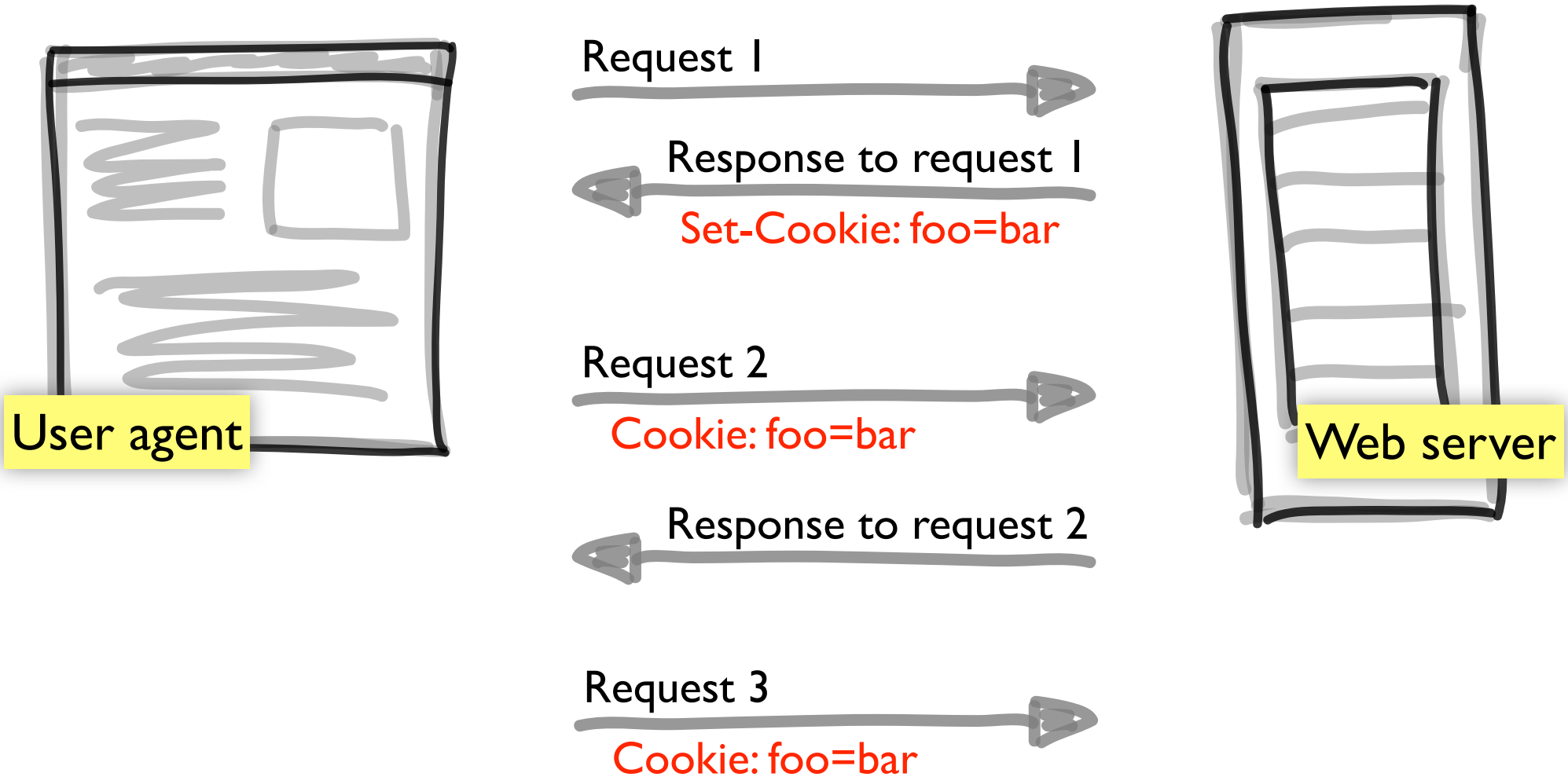
Information in HTTP headers in red

Cookies in HTTP requests and responses



Information in HTTP headers in red

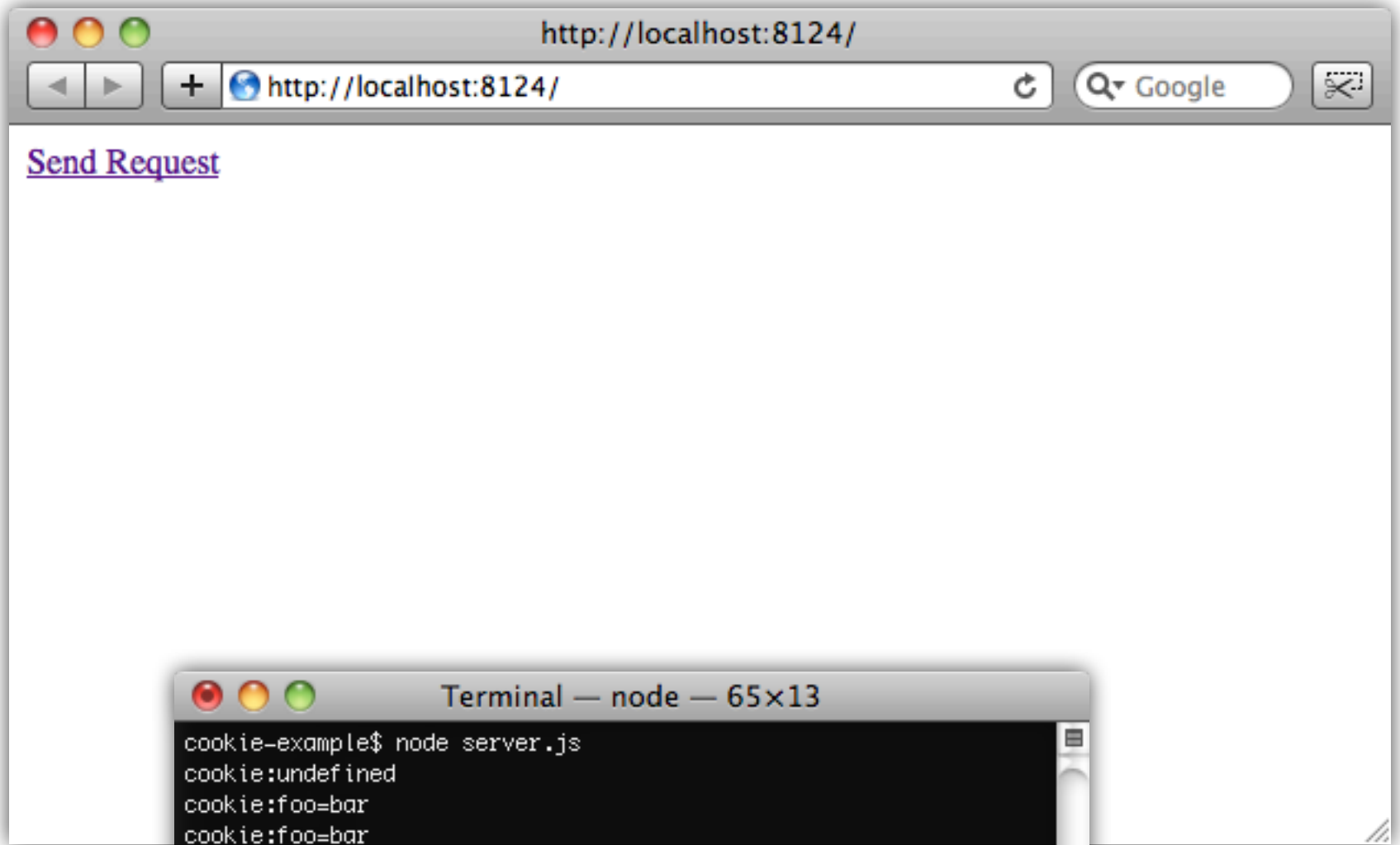
Cookies in HTTP requests and responses

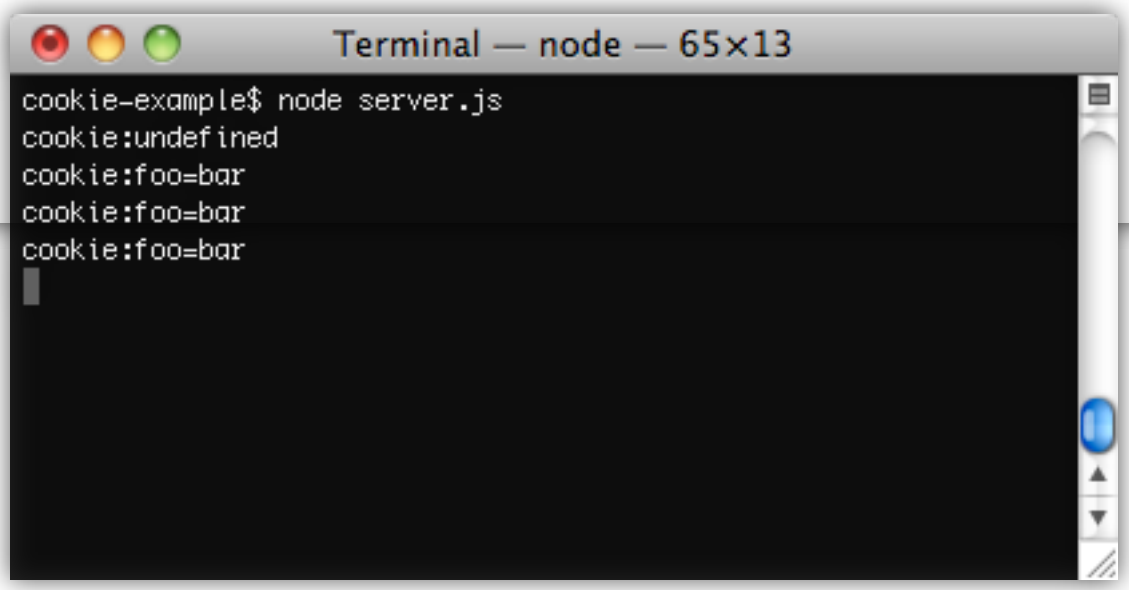


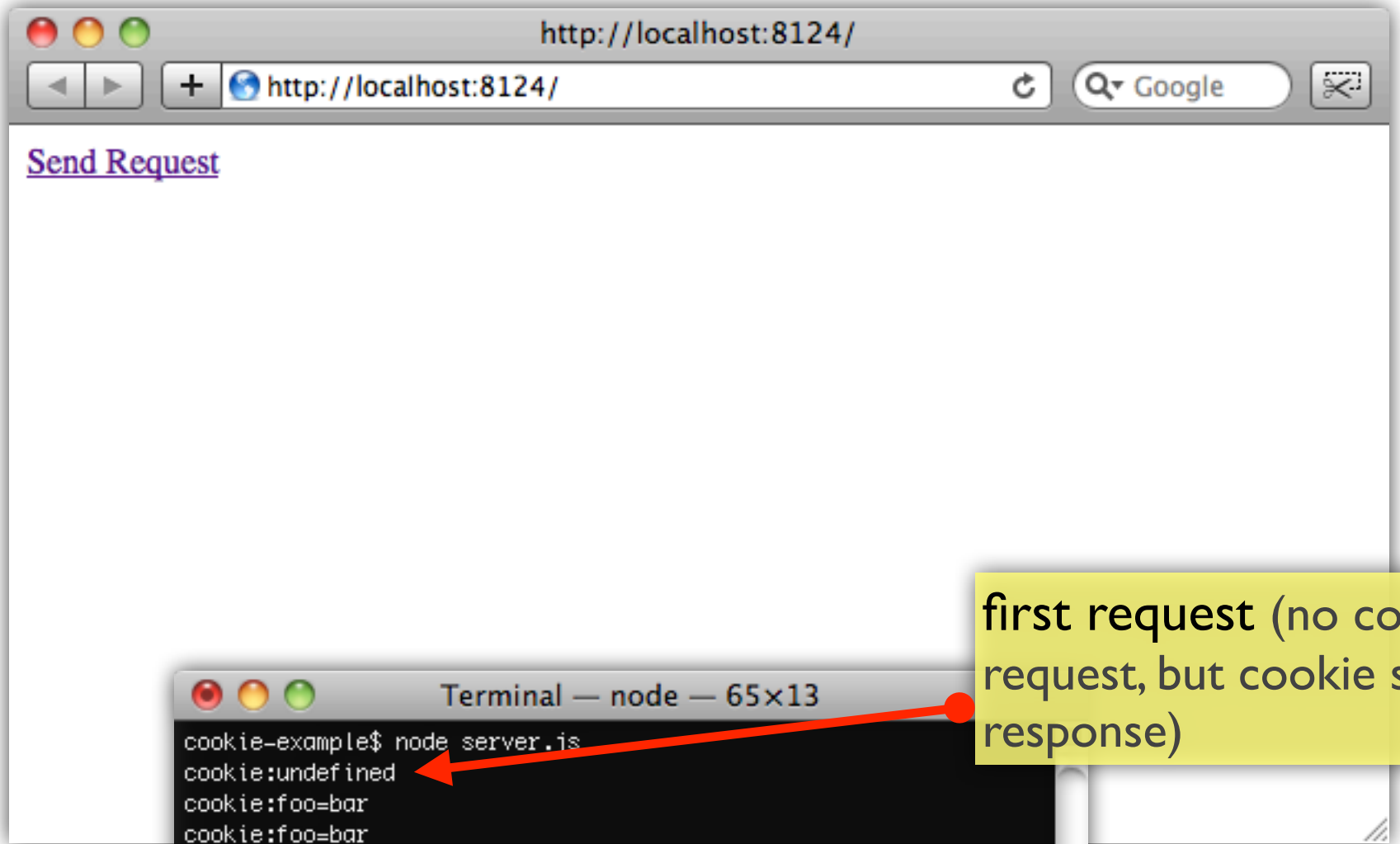
Information in HTTP headers in red

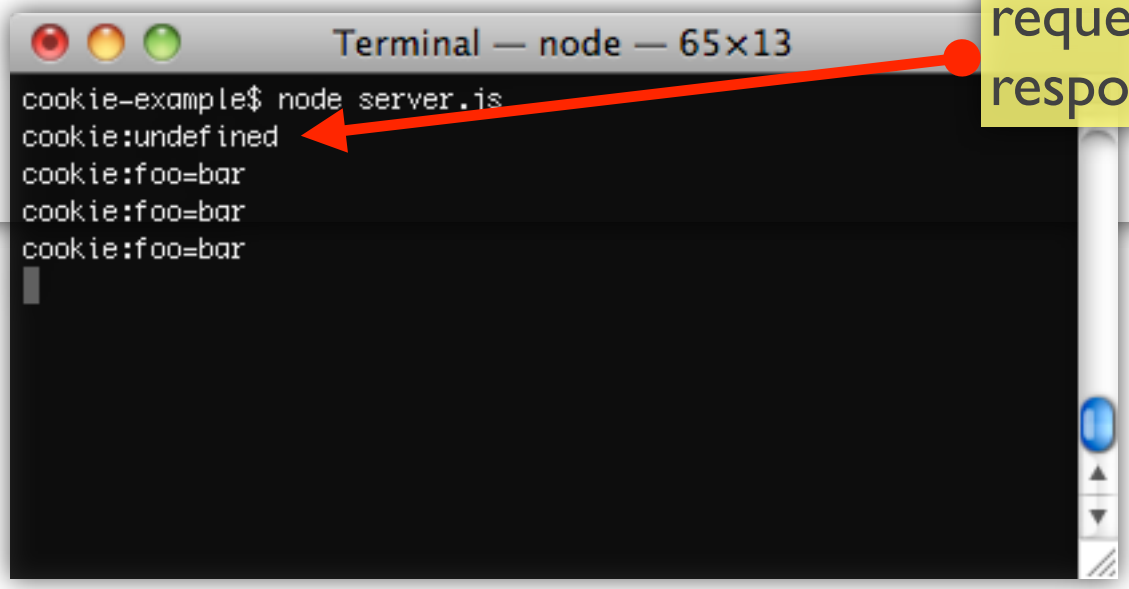
```
http.createServer(function (request, response) {
    console.log('cookie:' + request.headers['cookie'])

    response.writeHead(200, {
        'content-type': 'text/html',
        'set-cookie': 'foo=bar',
    })

    response.write('<!DOCTYPE html>\n')
    response.write('<html><body>\n')
    response.write('<a href="/">Send Request</a>\n')
    response.write('</body></html>\n')
    response.end()

}).listen(8124)
```


http://localhost:8124/
http://localhost:8124/
Google
Send Request

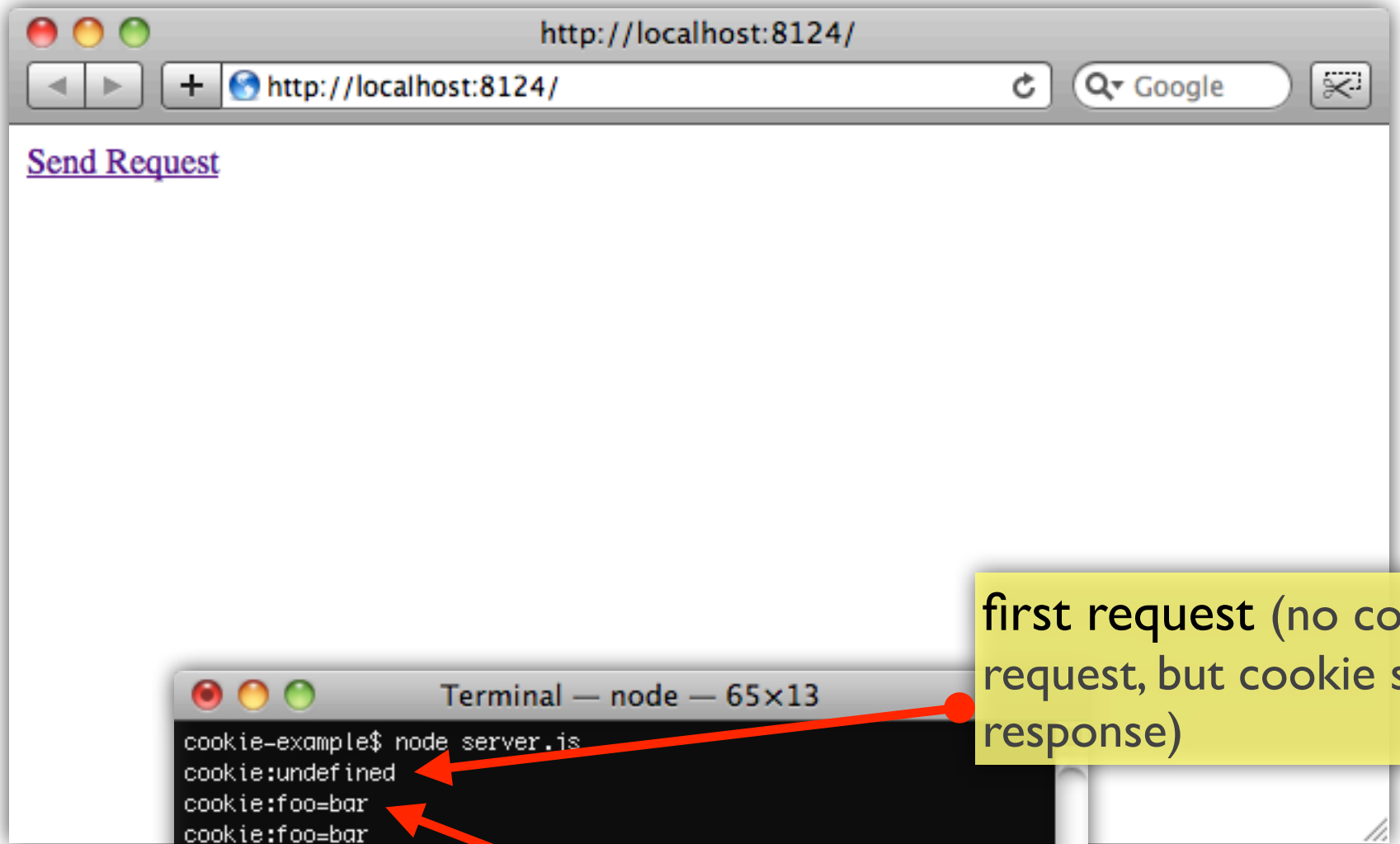

http://localhost:8124/
http://localhost:8124/
Google
Send Request

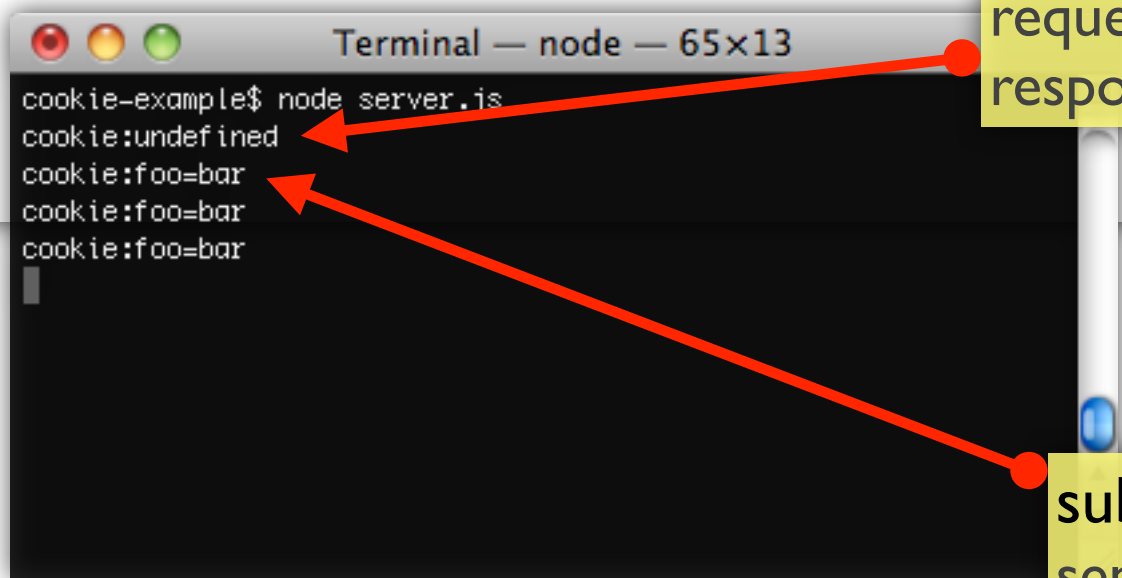

Terminal — node — 65×13
cookie-example$ node server.js
cookie:undefined
cookie:foo=bar
cookie:foo=bar
cookie:foo=bar





first request (no cookie in request, but cookie sent with response)





first request (no cookie in request, but cookie sent with response)

subsequent requests (cookie sent with each request)

```
http.createServer(function (request, response) {
    console.log('cookie:' + request.headers['cookie'])

    response.writeHead(200, {
        'content-type': 'text/html',
        'set-cookie': 'foo=bar',
    })

    response.write('<!DOCTYPE html>\n')
    response.write('<html><body>\n')
    response.write('<a href="/">Send Request</a>\n')
    response.write('</body></html>\n')
    response.end()

}).listen(8124)
```

HTTP Cookies

Photo from: http://www.flickr.com/photos/mrsmagic/1117398599/

Overview

1. Cookies: the simple version
2. Using cookies for managing sessions
3. Cookies: more details

HTTP, State and Cookies

HTTP is a stateless protocol. This means that by default no information or status about each user agent is saved across multiple request.

Cookies give us a way to work around this problem by introducing a notion of a **session** (which might begin when the user logs in and end when the user logs out, or some amount of time passes).

Using Cookies to Manage **Sessions**

- Information about the session is stored on the server (possibly in memory or in the database or ...)
- The server generates a unique **id** for each session (called the session id)
- The server uses the HTTP **Set-Cookie** header to set a cookie with that unique session id
- The browser sends the unique session id with every subsequent HTTP request to the server using the **Cookie** HTTP header
- The server uses the session id to get the session information

Session Notes

In general only the session id is stored in the cookie. Other session information is stored on the server. This might be saved in memory or a database.

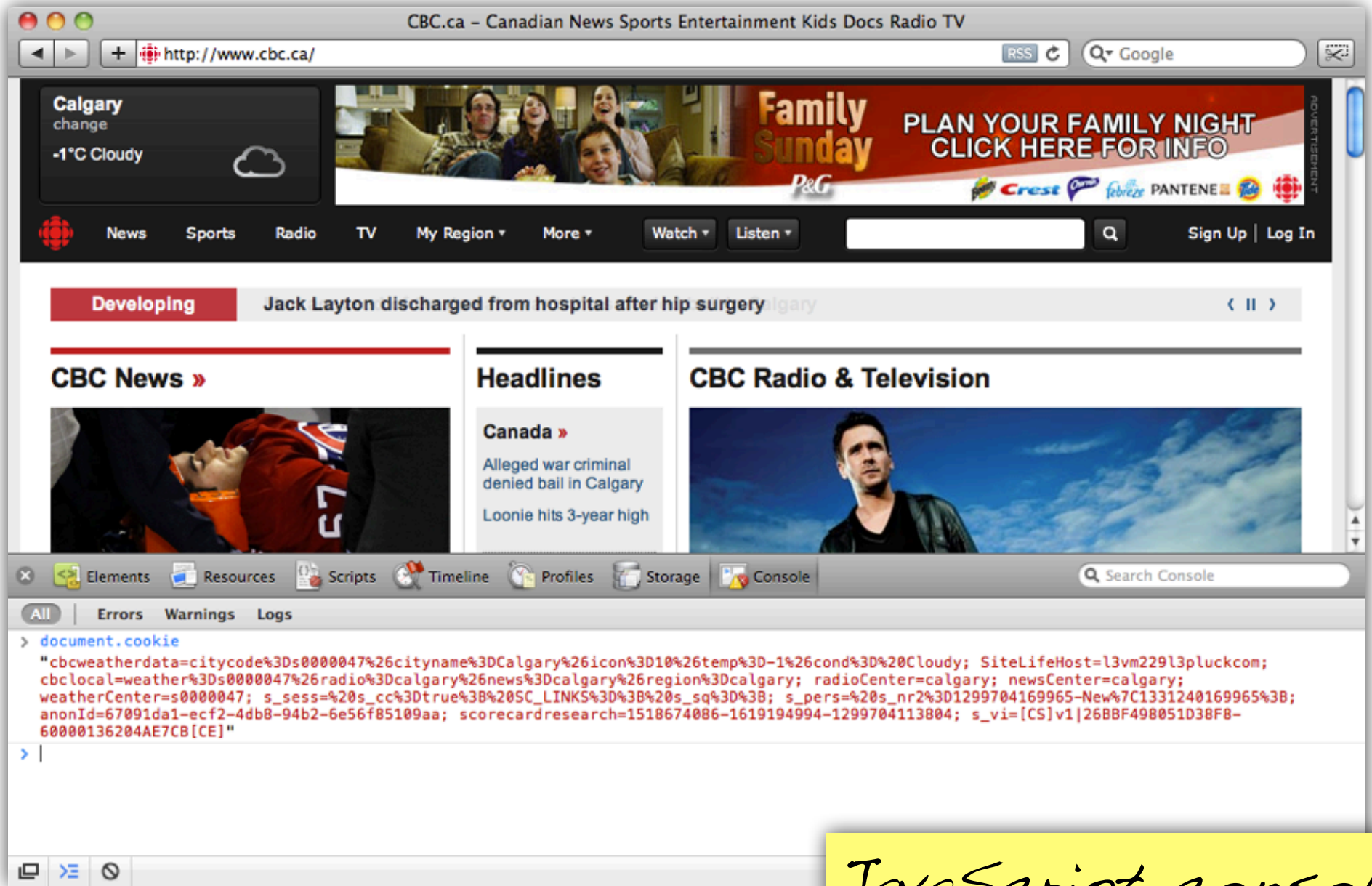
(More on security later in the term, time permitting.)

HTTP Cookies

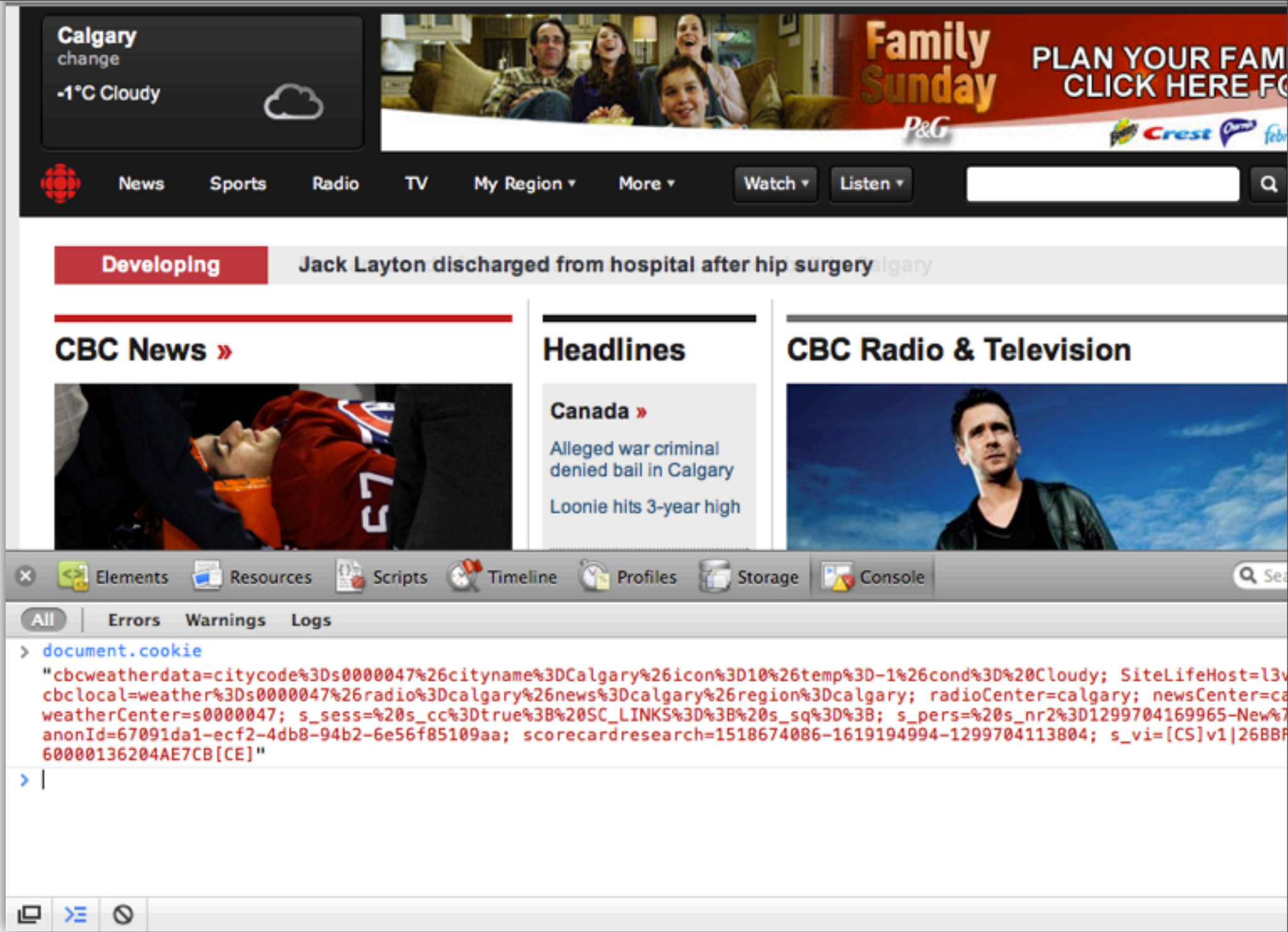
Photo from: http://www.flickr.com/photos/mrsmagic/1117398599/

Overview

1. Cookies: the simple version
2. Using cookies for managing sessions
3. Cookies: more details



JavaScript console


Calgary
change
-1°C Cloudy
Family Sunday
PLAN YOUR FAM
CLICK HERE FO
P&G
Crest
News
Sports
Radio
TV
My Region
More
Watch
Listen
Developing
Jack Layton discharged from hospital after hip surgery
CBC News »
Headlines
Canada »
Alleged war criminal denied bail in Calgary
Loonie hits 3-year high
CBC Radio & Television
Elements
Resources
Scripts
Timeline
Profiles
Storage
Console
All
Errors
Warnings
Logs
> document.cookie
"cbcweatherdata=citycode%3Ds0000047%26cityname%3DCalgary%26icon%3D10%26temp%3D-1%26cond%3D%20Cloudy; SiteLifeHost=l3v
cbclocal=weather%3Ds0000047%26radio%3Dcalgary%26news%3Dcalgary%26region%3Dcalgary; radioCenter=calgary; newsCenter=ca
weatherCenter=s0000047; s_sess=%20s_cc%3Dtrue%3B%20SC_LINKS%3D%3B%20s_sq%3D%3B; s_pers=%20s_nr2%3D1299704169965-New%7
anonId=67091da1-ecf2-4db8-94b2-6e56f85109aa; scorecardresearch=1518674086-1619194994-1299704113804; s_vi=[CS]v1|26BBF
60000136204AE7CB[CE]"

Cookie Attributes

In addition to a name-value pair, cookies set by a server can have several attributes (note: only the name-value pair is sent back by the user-agent to the server).

- **expires or max-age** - tells the user agent how long to keep the cookie around. If no expiration is provided, the cookie will go away after the browser is closed.
- **domain + path** - basically define a scope for the cookie and the user agent only sends the cookie back for requests with matching domains and paths.
- **secure** - tells user agent to send cookie only for secure connections.
- **httpOnly** - tells user agent to only use cookie in context of the HTTP protocol.

Example Cookies

Google

PREF=ID=5567078546fd61cb:FF=0:TM=1299540154:LM=1299540154:S=nTVA9iOA0dOO1OSf; expires=Wed, 06-Mar-2013 23:22:34 GMT; path=/; domain=.google.com

Facebook

datr=RWI1TSjZaENsh_LdACjUw1UL; expires=Wed, 06-Mar-2013 23:24:53 GMT; path=/; domain=.facebook.com; httponly

Example Cookies

Google

PREF=ID=5567078546fd61cb:FF=0:TM=1299540154:LM=1299540154:S=nTVA9iOA0dOO1OSf; expires=Wed, 06-Mar-2013 23:22:34 GMT; path=/; domain=.google.com

Facebook

datr=RWI1TSjZaENsh_LdACjUw1UL; expires=Wed, 06-Mar-2013 23:24:53 GMT; path=/; domain=.facebook.com; httponly

semi-colons divide the cookie attributes

Example Cookies

Google

PREF=ID=5567078546fd61cb:FF=0:TM=1299540154:LM=1299540154:S=nTVA9iOA0dOO1OSf; expires=Wed, 06-Mar-2013 23:22:34 GMT; path=/; domain=.google.com

Facebook

datr=RWI1TSjZaENsh_LdACjUw1UL; expires=Wed, 06-Mar-2013 23:24:53 GMT; path=/; domain=.facebook.com; httponly

name value pair (the main cookie data)

Example Cookies

Google

PREF=ID=5567078546fd61cb:FF=0:TM=1299540154:LM=1299540154:S=nTVA9iOA0dOO1OSf; expires=Wed, 06-Mar-2013 23:22:34 GMT; path=/; domain=.google.com

Facebook

datr=RWI1TSjZaENsh_LdACjUw1UL; expires=Wed, 06-Mar-2013 23:24:53 GMT; path=/; domain=.facebook.com; httponly

expiration (tells user agent how long to keep cookie)

Example Cookies

Google

PREF=ID=5567078546fd61cb:FF=0:TM=1299540154:LM=1299540154:S=nTVA9iOA0dOO1OSf; expires=Wed, 06-Mar-2013 23:22:34 GMT; path=/; domain=.google.com

Facebook

datr=RWI1TSjZaENsh_LdACjUw1UL; expires=Wed, 06-Mar-2013 23:24:53 GMT; path=/; domain=.facebook.com; httponly

path and domain tell the user agent when to send cookie with a request

Example Cookies

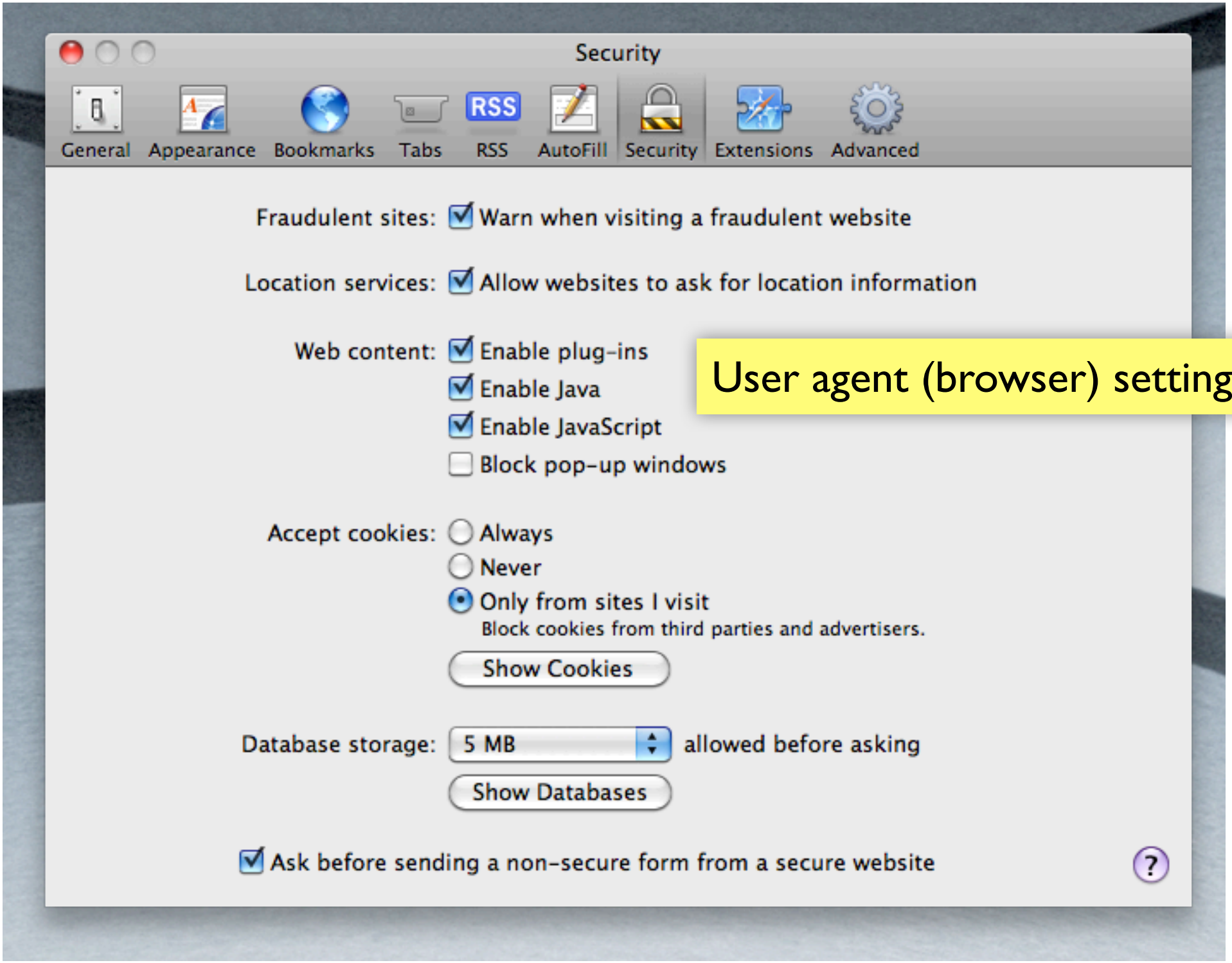
Google

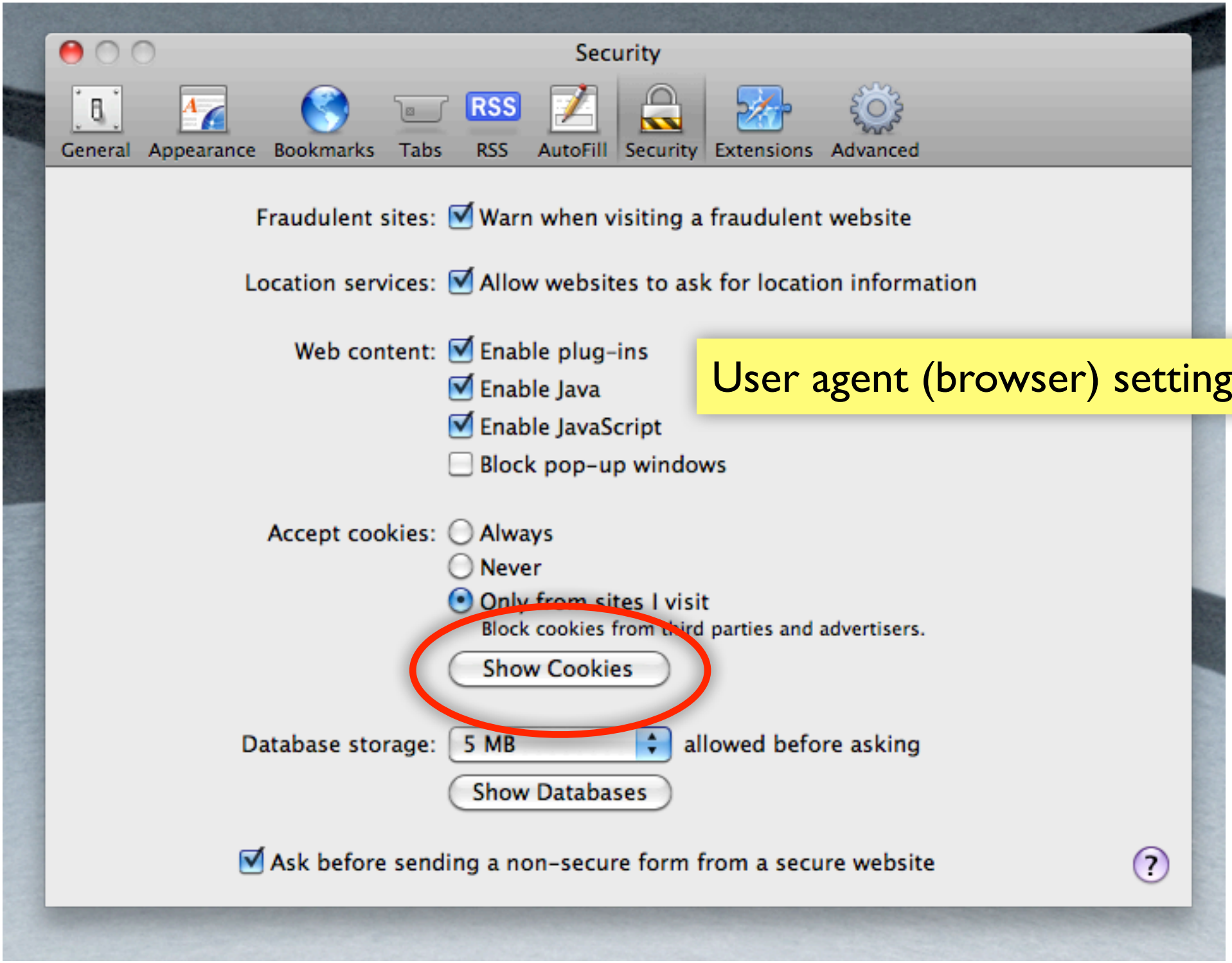
PREF=ID=5567078546fd61cb:FF=0:TM=1299540154:LM=1299540154:S=nTVA9iOA0dOO1OSf; expires=Wed, 06-Mar-2013 23:22:34 GMT; path=/; domain=.google.com

Facebook

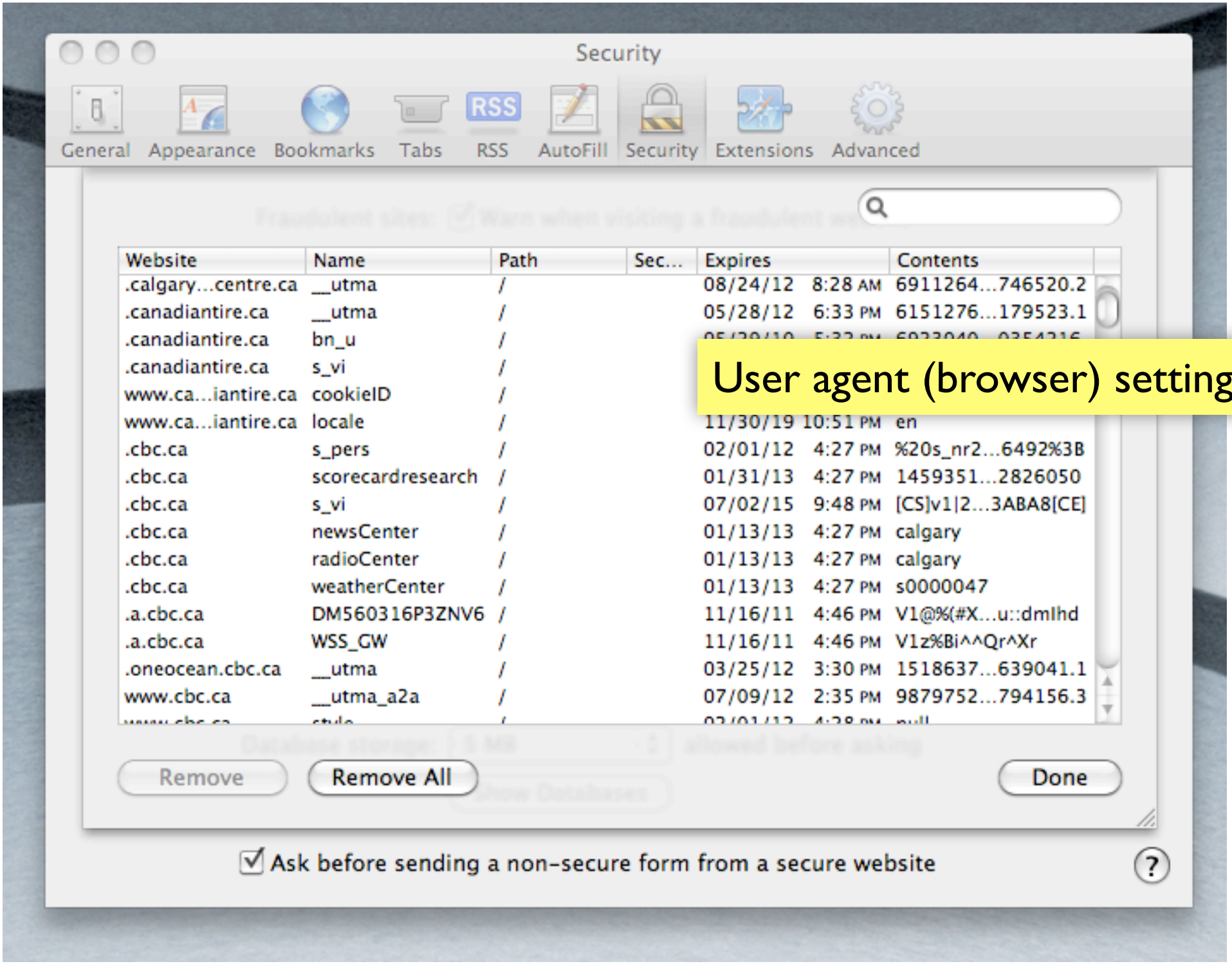
datr=RWI1TSjZaENsh_LdACjUw1UL; expires=Wed, 06-Mar-2013 23:24:53 GMT; path=/; domain=.facebook.com; httponly

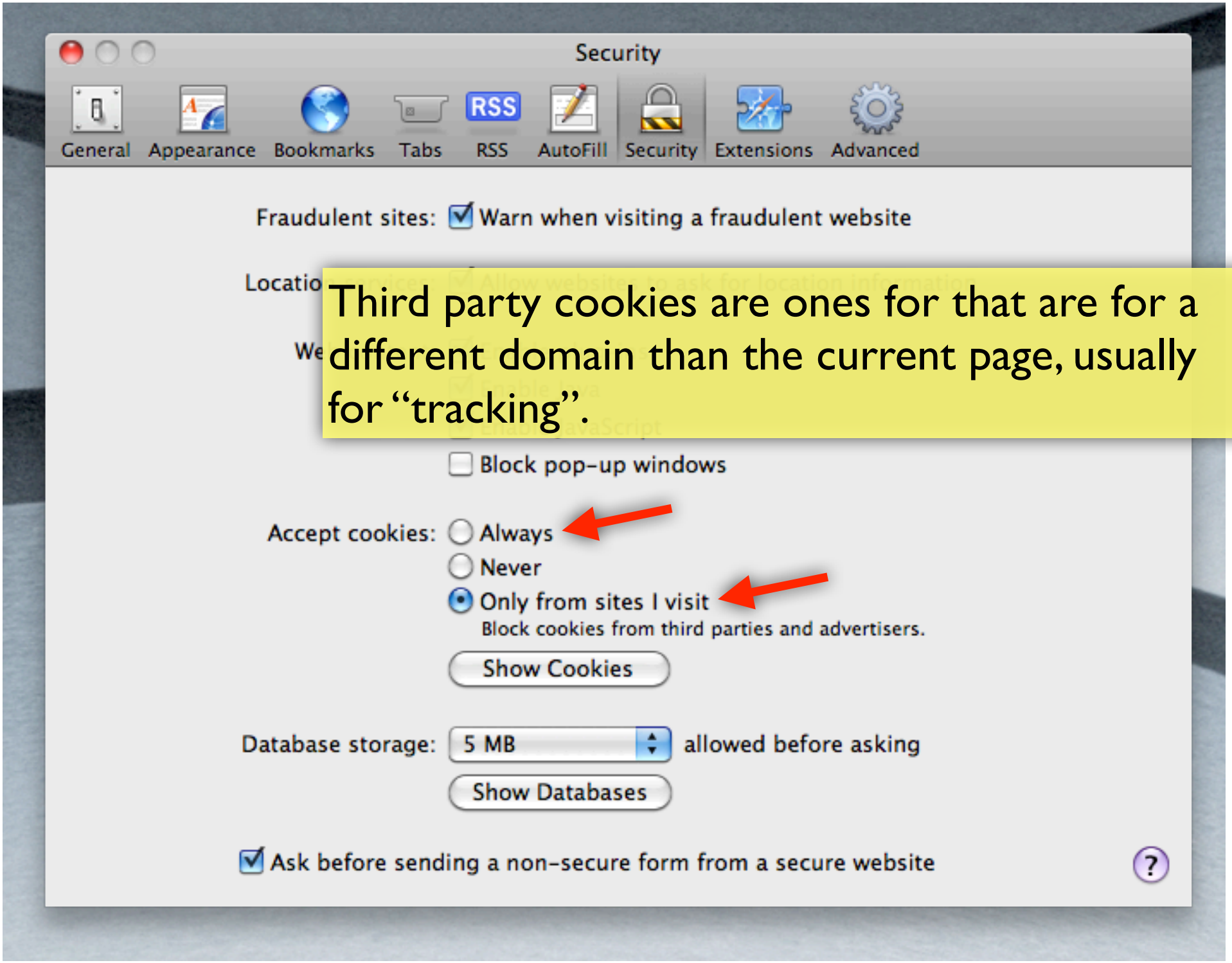
httponly tells user agent to only the send cookie with http requests (i.e., so it is not accessible to scripts)


Security
General
Appearance
Bookmarks
Tabs
RSS
AutoFill
Security
Extensions
Advanced
Fraudulent sites:
Warn when visiting a fraudulent website
Location services:
Allow websites to ask for location information
Web content:
Enable plug-ins
Enable Java
Enable JavaScript
Block pop-up windows
User agent (browser) settings
Accept cookies:
Always
Never
Only from sites I visit
Block cookies from third parties and advertisers.
Show Cookies
Database storage:
5 MB
allowed before asking
Show Databases
Ask before sending a non-secure form from a secure website



User agent (browser) settings


Security
General
Appearance
Bookmarks
Tabs
RSS
AutoFill
Security
Extensions
Advanced
Website
Name
Path
Sec...
Expires
Contents
.calgary...centre.ca __utma / 08/24/12 8:28 AM 6911264...746520.2
.canadiantire.ca __utma / 05/28/12 6:33 PM 6151276...179523.1
.canadiantire.ca bn_u /
.canadiantire.ca s_vi /
www.ca...iantire.ca cookieID /
www.ca...iantire.ca locale / 11/30/19 10:51 PM en
.cbc.ca s_pers / 02/01/12 4:27 PM %20s_nr2...6492%3B
.cbc.ca scorecardresearch / 01/31/13 4:27 PM 1459351...2826050
.cbc.ca s_vi / 07/02/15 9:48 PM [CS]v1|2...3ABA8[CE]
.cbc.ca newsCenter / 01/13/13 4:27 PM calgary
.cbc.ca radioCenter / 01/13/13 4:27 PM calgary
.cbc.ca weatherCenter / 01/13/13 4:27 PM s0000047
.a.cbc.ca DM560316P3ZNV6 / 11/16/11 4:46 PM V1@%(#X...u::dmlhd
.a.cbc.ca WSS_GW / 11/16/11 4:46 PM V1z%Bi^^Qr^Xr
.oneocean.cbc.ca __utma / 03/25/12 3:30 PM 1518637...639041.1
www.cbc.ca __utma_a2a / 07/09/12 2:35 PM 9879752...794156.3
Remove
Remove All
Done
Ask before sending a non-secure form from a secure website
User agent (browser) settings


Security
General
Appearance
Bookmarks
Tabs
RSS
AutoFill
Security
Extensions
Advanced
Fraudulent sites:
Warn when visiting a fraudulent website
Third party cookies are ones for that are for a different domain than the current page, usually for "tracking".
Block pop-up windows
Accept cookies:
Always
Never
Only from sites I visit
Block cookies from third parties and advertisers.
Show Cookies
Database storage:
5 MB
allowed before asking
Show Databases
Ask before sending a non-secure form from a secure website