

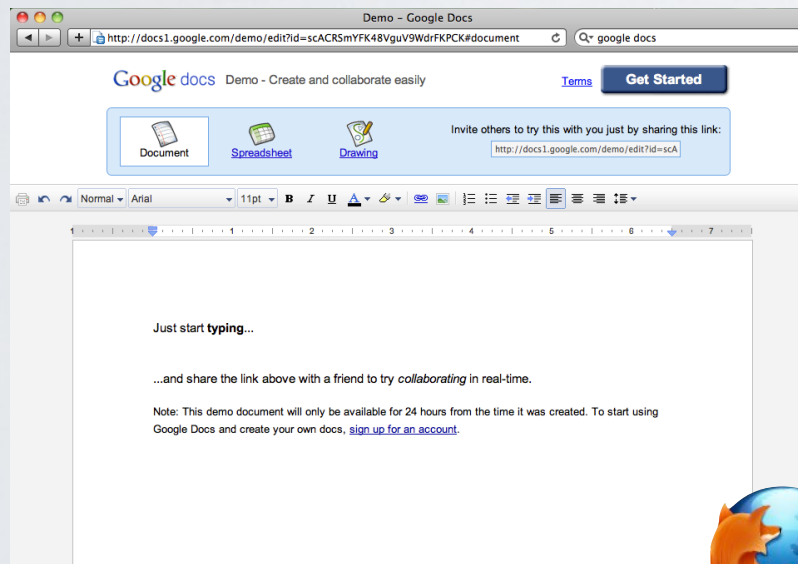
Cookies and Sessions

Thierry

Security assumptions

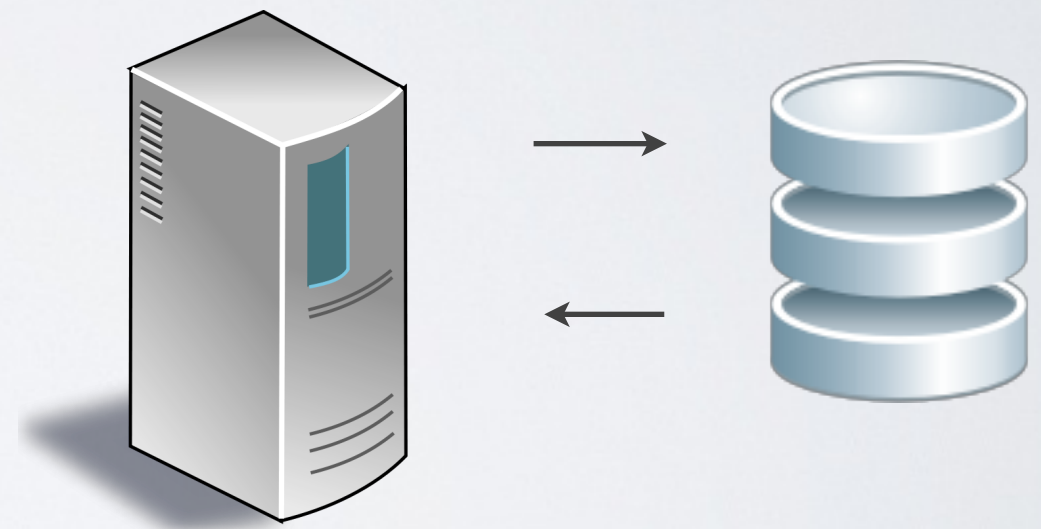
You have **absolutely no control** on the client

Client Side



Web Browser

Server Side



Web Server

Database


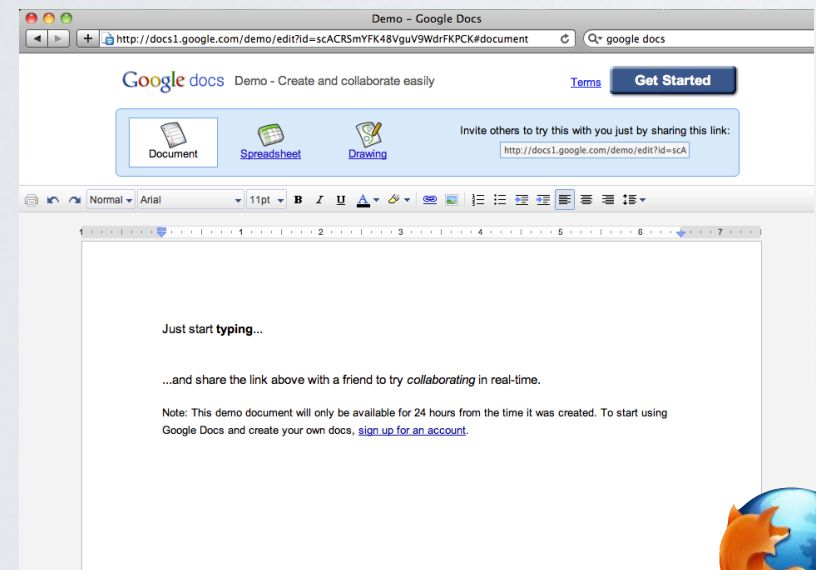
Cookies

The big picture

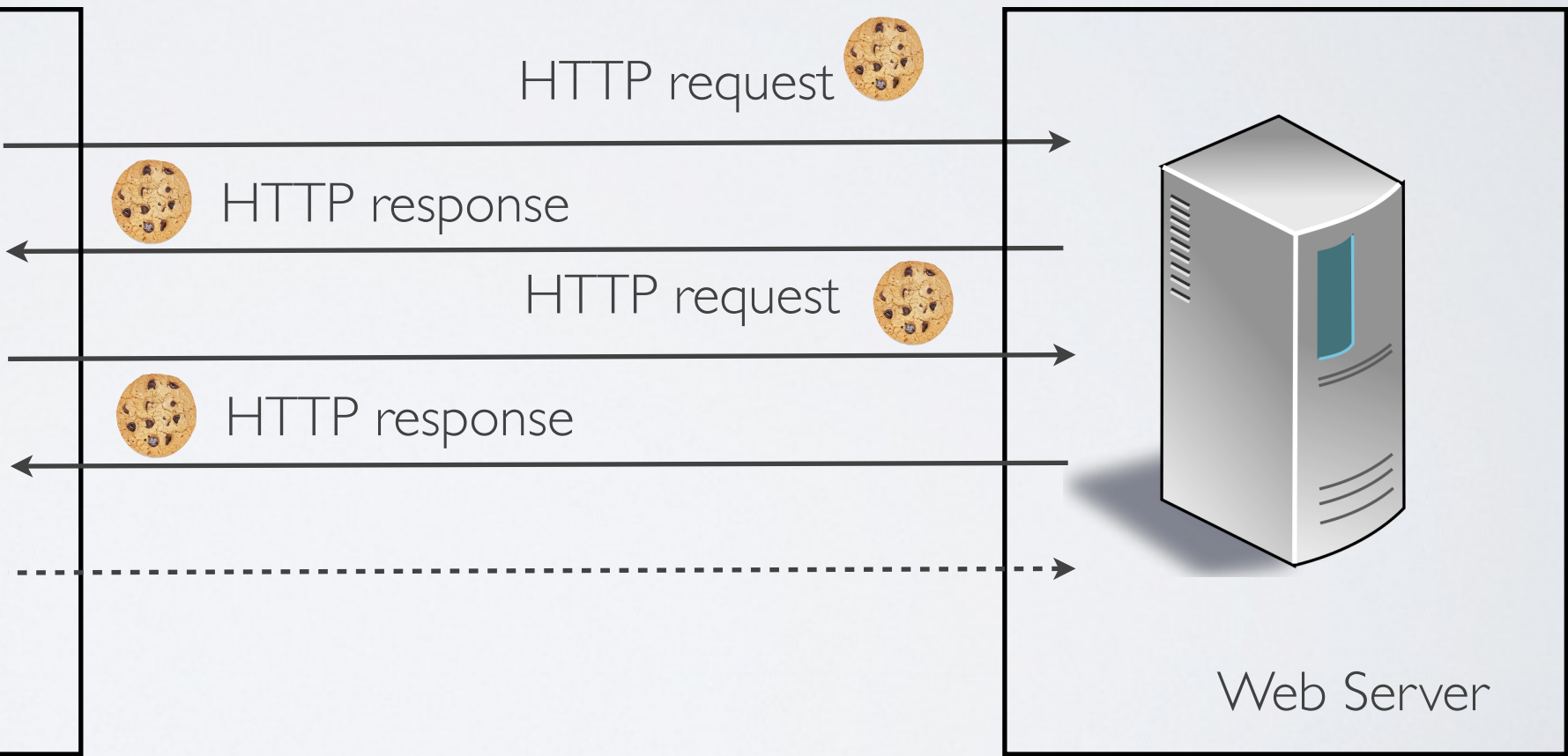
key/value pairs data

Client Side

Server Side



Web Browser



Cookies

Cookies are **key/value pairs** sent back and forth between the browser and the server in HTTP request and response

Anatomy of a Cookie

- Text data (Up to 4kb)
- May (or may not) have an expiration date
- Bound to a domain name and a path
- May have security flags (coming later)
- Can be manipulated from the client **and** the server

Manipulating cookies

A cookie can be modified

- on the **server** side
express middleware : **cookie**
- on the **client** side (unless `HttpOnly` flag is set)
javascript : **Document.cookie**


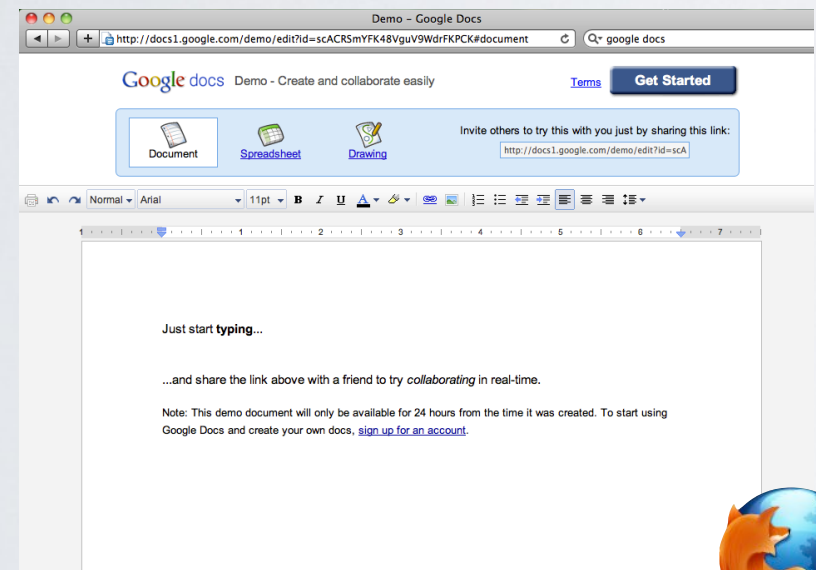
What cookies are useful for?

- Shopping cart
- Browsing preferences
- User authentication
- Tracking and advertisement

Sessions

The big picture

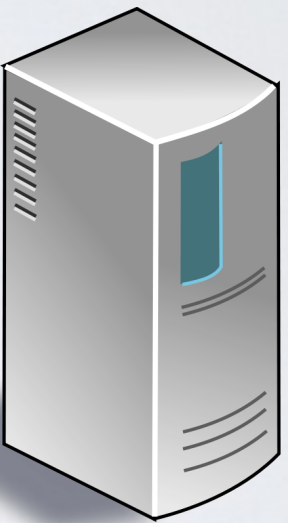
Client Side



Web Browser

session id

Server Side



Web Server

key/value pairs data

HTTP request

HTTP response

HTTP request

HTTP response

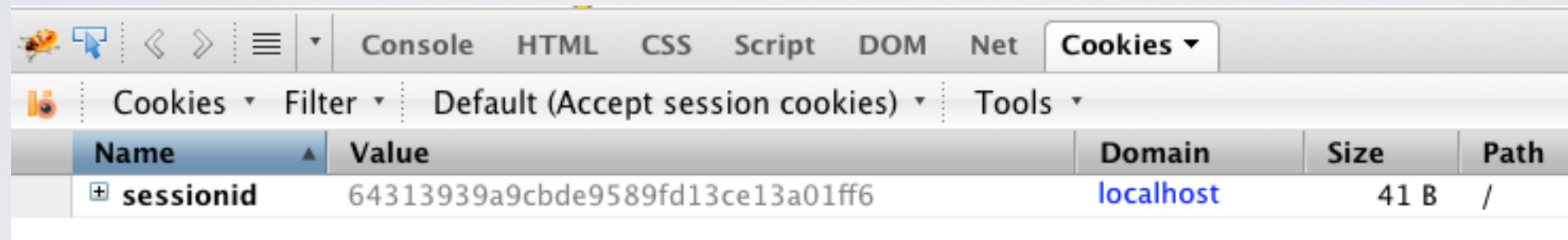
The concept of session

- There is a **session id** (aka token) between the browser and the web application
- This session id should be **unique** and **unforgeable** (usually a long random number or a hash)
- This session id is bind to **key/value pairs data**

Where sessions values are stored

- **Session ID** is stored in a cookie
- **Session key/value pairs** are stored on the server

Hacking sessions



Name	Value	Domain	Size	Path
<input type="checkbox"/> sessionid	64313939a9cbde9589fd13ce13a01ff6	localhost	41 B	/

The user can **create, modify, delete** the session ID in the cookie

But **cannot access** the key/value pairs stored on the server