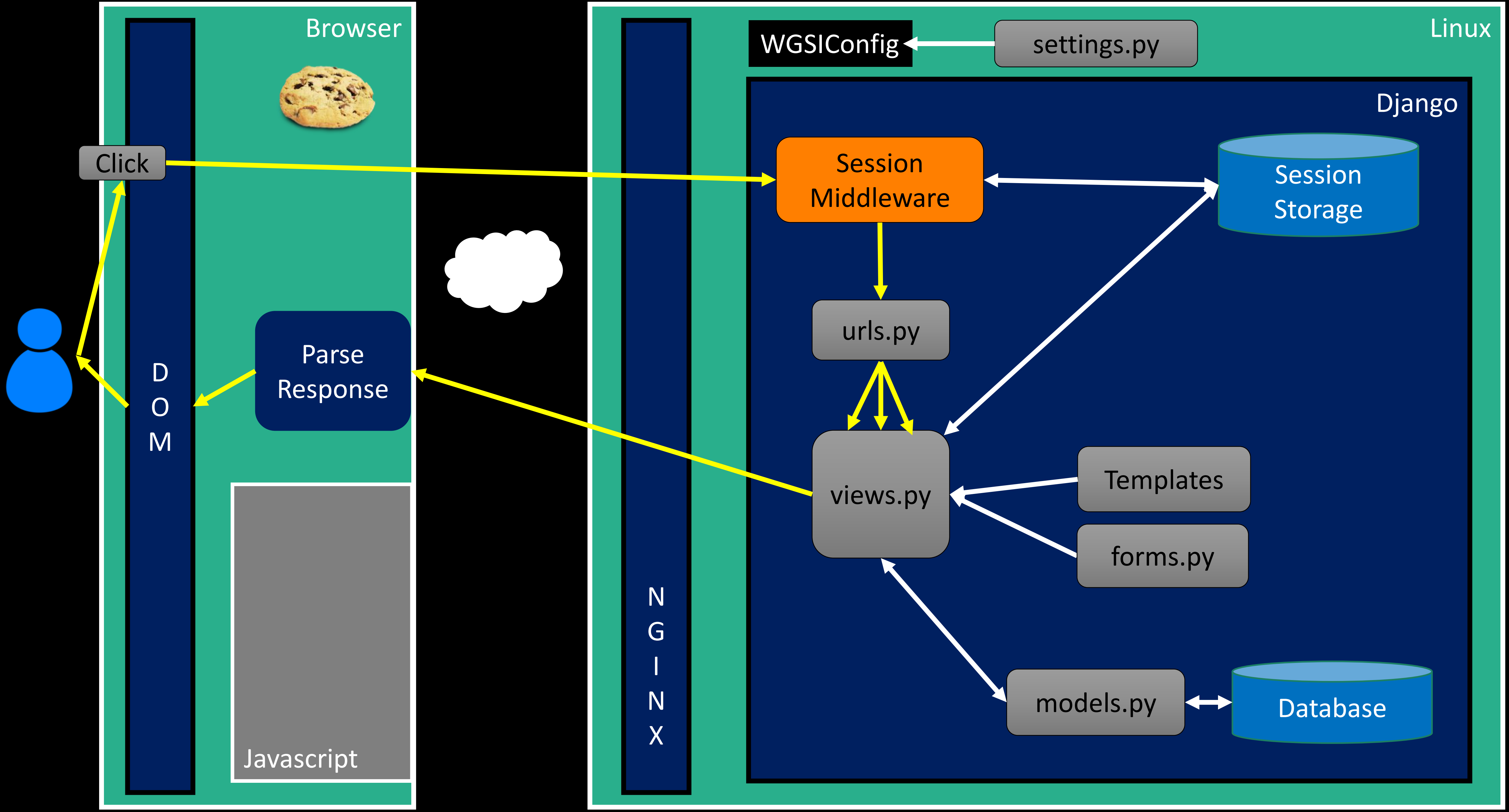
Charles Severance

www.dj4e.com

Cookies and Sessions

https://samples.dj4e.com/session/


Browser
Click
D O M
Parse Response
Javascript
Linux
N G I N X
WGSIConfig
settings.py
Django
Session Middleware
Session Storage
urls.py
views.py
Templates
forms.py
models.py
Database

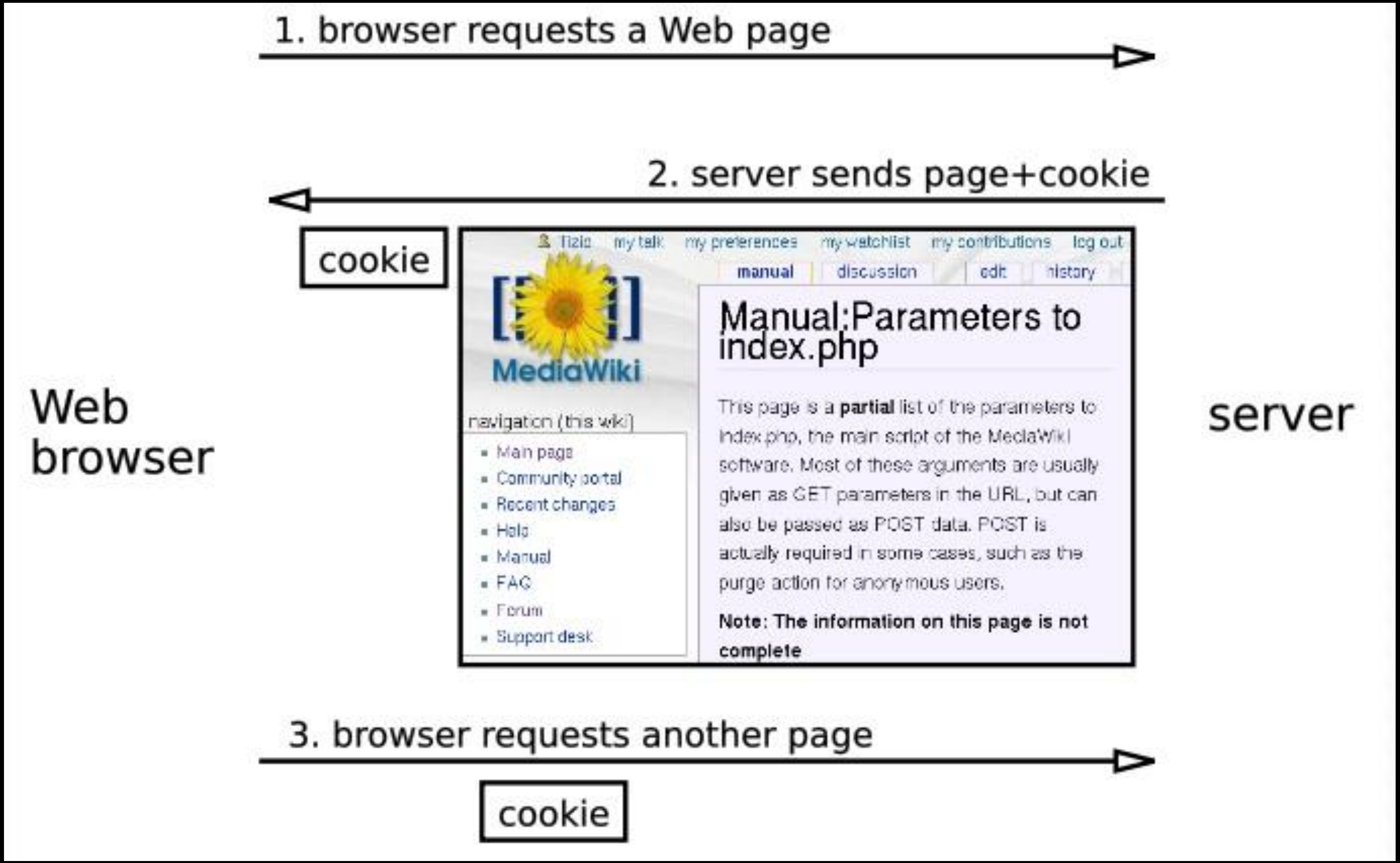
Multi-User / Multi-Browser

- When a server is interacting with many different browsers at the same time, the server needs to know *which* browser a particular request came from.
- Request / Response initially was stateless - all browsers looked identical . This was really bad and did not last very long at all.

Web Cookies to the Rescue

Technically, cookies are arbitrary pieces of data chosen by the Web server and sent to the browser. The browser returns them unchanged to the server, introducing a state (memory of previous events) into otherwise stateless HTTP transactions. Without cookies, each retrieval of a Web page or component of a Web page is an isolated event, mostly unrelated to all other views of the pages of the same site.

http://en.wikipedia.org/wiki/HTTP_cookie



Manual:Parameters to index.php

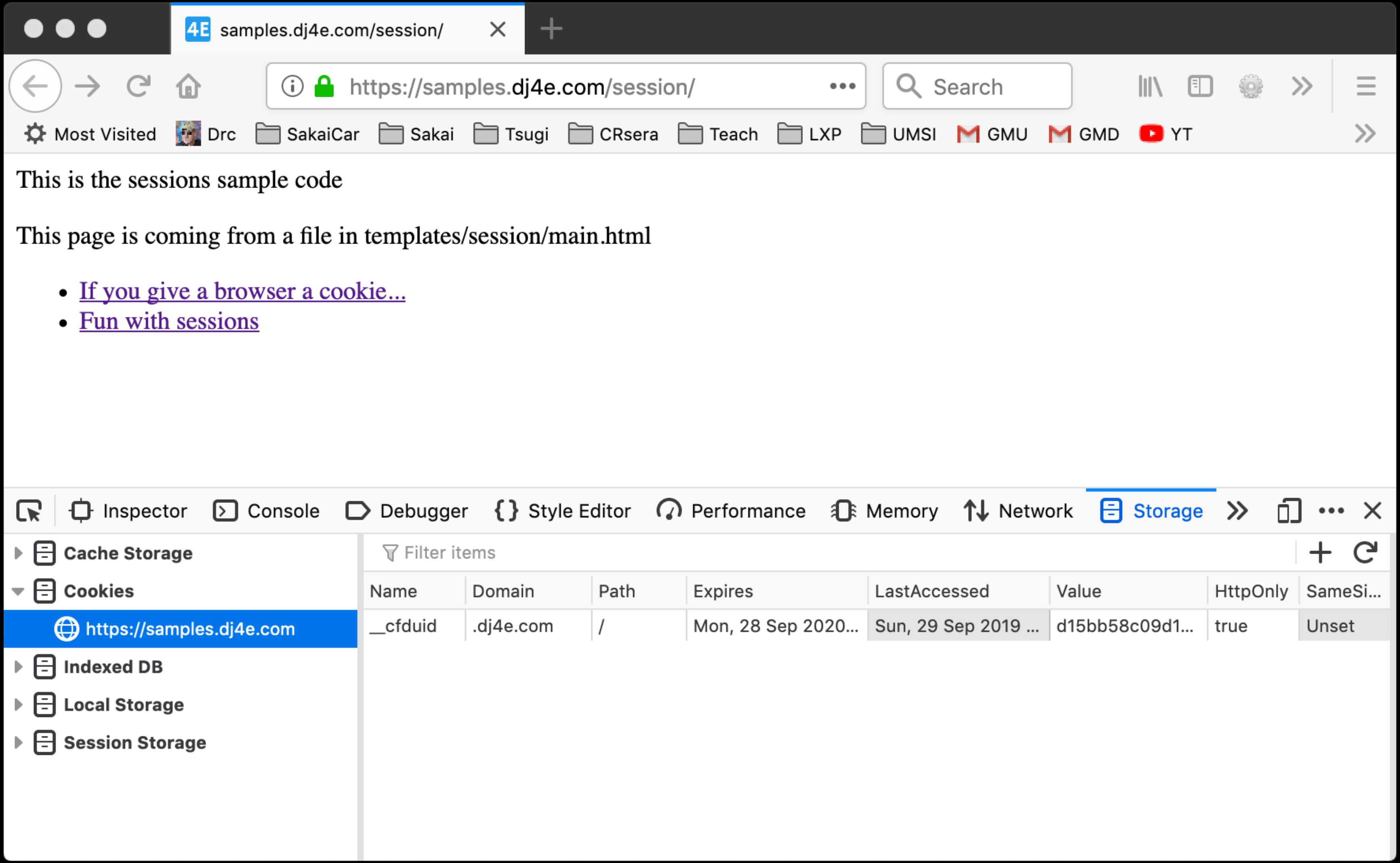
This page is a **partial** list of the parameters to index.php, the main script of the MediaWiki software. Most of these arguments are usually given as GET parameters in the URL, but can also be passed as POST data. POST is actually required in some cases, such as the purge action for anonymous users.

Note: The information on this page is not complete

http://en.wikipedia.org/wiki/HTTP_cookie

Cookies In the Browser

- Cookies are marked as to the web addresses they come from. The browser only sends back cookies that were originally set by the same web server.
- Cookies have an expiration date. Some last for years, others are short-term and go away as soon as the browser is closed



This is the sessions sample code

This page is coming from a file in templates/session/main.html

- If you give a browser a cookie...
- Fun with sessions

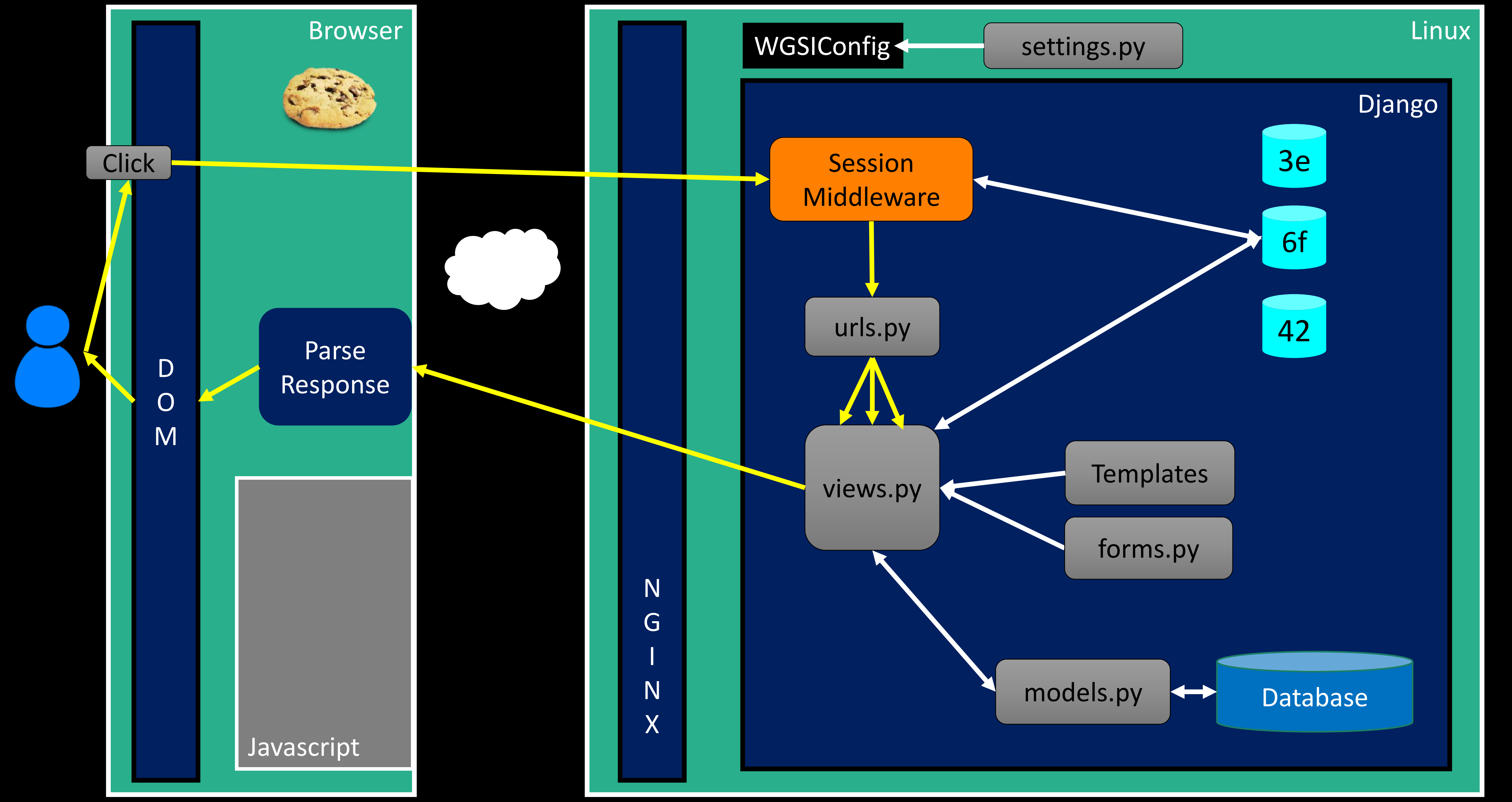
Name	Domain	Path	Expires	LastAccessed	Value	HttpOnly	SameSi...
__cfduid	.dj4e.com	/	Mon, 28 Sep 2020...	Sun, 29 Sep 2019 ...	d15bb58c09d1...	true	Unset

https://samples.dj4e.com/session/
https://github.com/csev/dj4e-samples/blob/master/session/home/views.py

Django Sessions

https://samples.dj4e.com/session/sessfun

https://github.com/csev/dj4e-samples/tree/master/session


Browser
Click
D O M
Parse Response
Javascript
Linux
N G I N X
WGSIConfig
settings.py
Django
Session Middleware
3e
6f
42
urls.py
views.py
Templates
forms.py
models.py
Database

In the Server - Sessions

- In most server applications, as soon as we start a session for a new (unmarked) browser we create a session.
- We set a session cookie to be stored in the browser, which indicates the session id in use – gives this browser a unique “mark”.
- The creation and destruction of sessions is handled by a Django middleware that we use in our applications.

Session Identifier

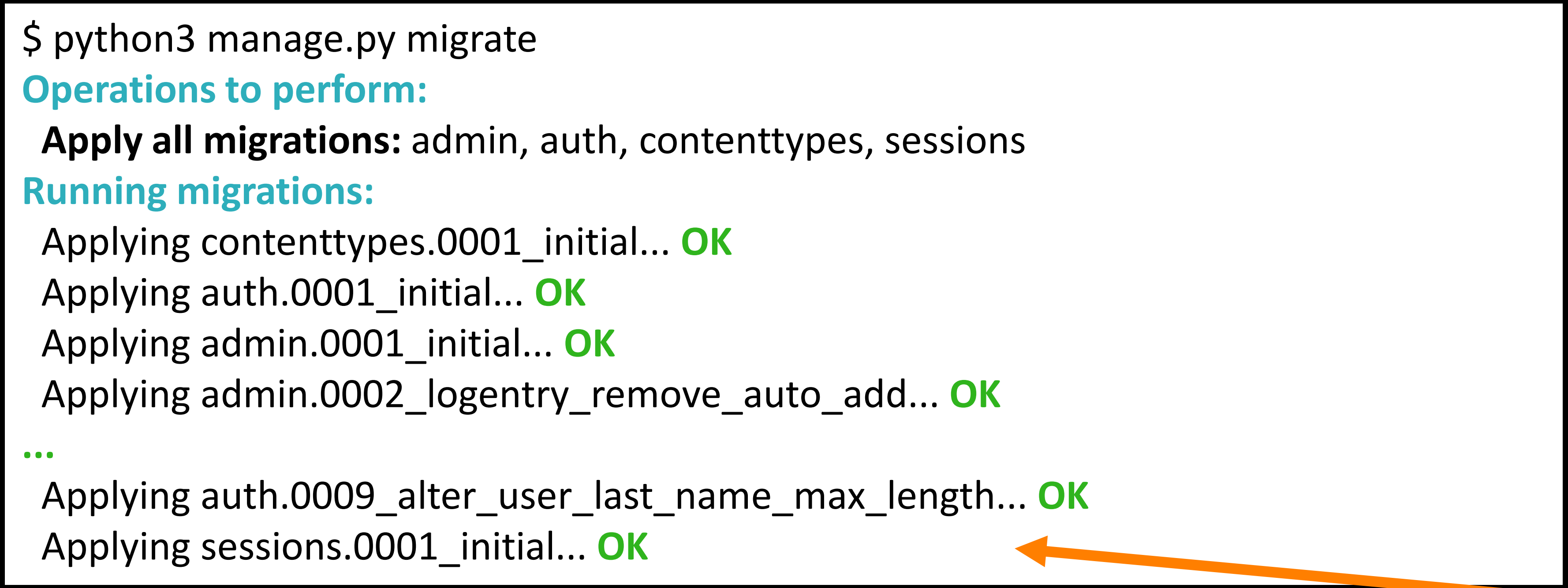
- A large, random number that we place in a browser cookie the first time we encounter a browser
- This number is used to pick from the many sessions that the server has active at any one time.
- Server software stores data in the session that it wants to have from one request to another from the same browser.
- Shopping cart or login information is stored in the session in the server.

Middleware

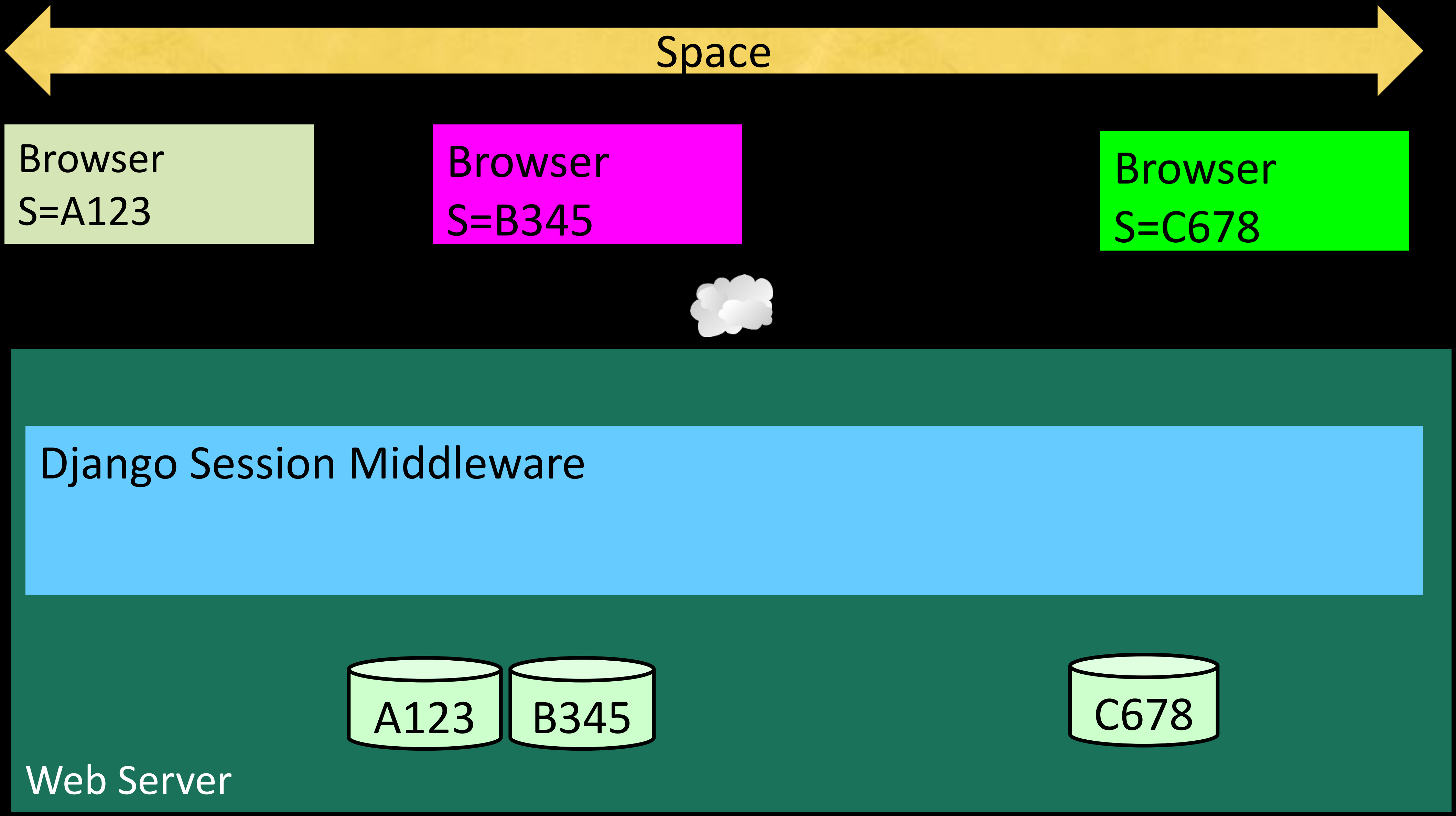
```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

https://github.com/csev/dj4e-samples/blob/master/dj4e-samples/settings.py

Default – Store Sessions in the Database



```
$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
...
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```


Space
Browser
S=A123
Browser
S=B345
Browser
S=C678
Django Session Middleware
A123
B345
C678
Web Server

Django Sessions

- The incoming request object has a request.session attribute that we can treat like a dictionary that persists from one request to the next request
- As long we have the session middleware enabled in settings.py and the database table, and the browser allows cookies, we just store and read request.session in our views and pretend it is "magic"

Additional Source Information

- Cookie Image: By brainloc on sxc.hu (Bob Smith) (stock.xchng) [CC BY 2.5 (http://creativecommons.org/licenses/by/2.5)], via Wikimedia Commons
- Portions of the text of these slides is adapted from the text www.djangoproject.org web site. Those slides which use text from that site have a reference to the original text on that site. Django is licensed under the three-clause BSD license.