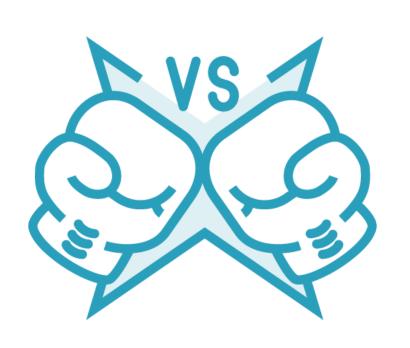
Django Cookies, Sessions and System Checks



Saravanan Dhandapani SOFTWARE ARCHITECT

@dsharu

Web Application Protocol



Stateless protocol

Cannot maintain previous history

Cookies

- Data stored in browser



Cookie Lifecycle

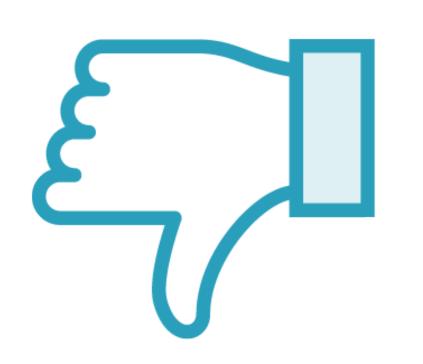




Django Session Framework



Disadvantages of Cookie

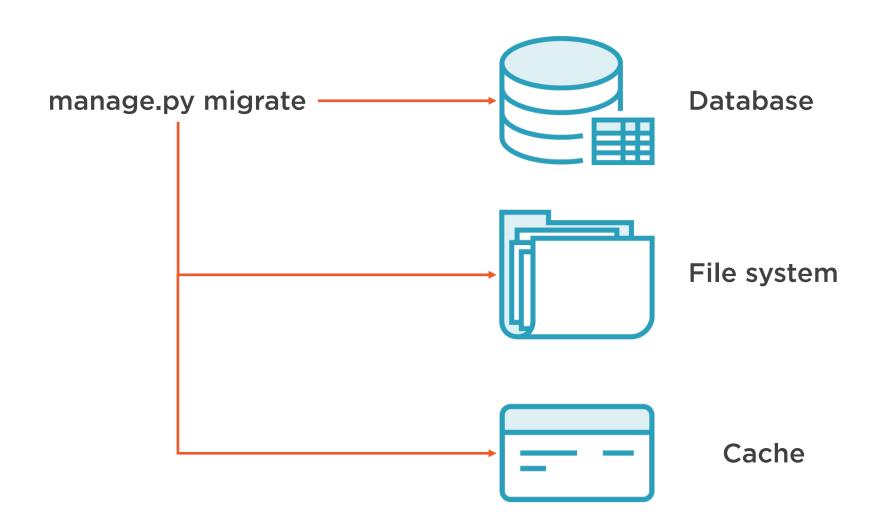


Client-side maintenance

User disabling cookies in their browser

Potential target for hackers

Session Storage





Memcached

```
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:8008',
     }
}
```



Cached Sessions

SESSION_ENGINE = "Django.contrib.sessions.backends.cache"



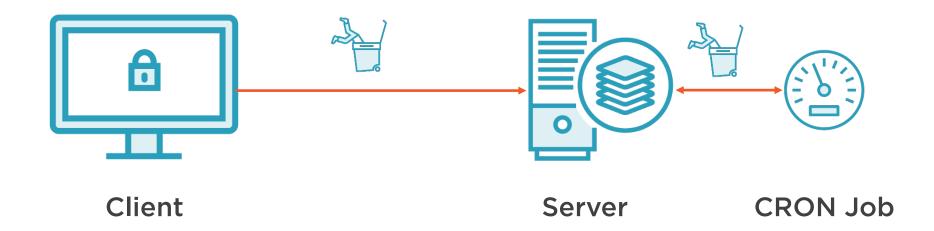
File-based Sessions

SESSION_ENGINE = "Django.contrib.sessions.backends.file"

SESSION_FILE_PATH = "<session_storage_directory>"



Session Purging





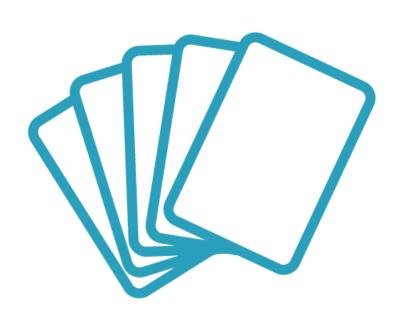
Demo



Django session management



Types of Sessions



Browser-length session

- Session_expire_at_browser_close = true

Persistent session

- Session_expire_at_browser_close = false
- Session_cookie_age expires



Django Messaging Framework



Django Messaging Framework



To display a flash message

- One-time notification message

Messages tagged with a level

- debug, info, success, warning and error

Fallback storage is default

Introduce "message_storage" to overwrite default



Demo



Django messaging framework



Demo



Django cookies



Django Convenience Features

