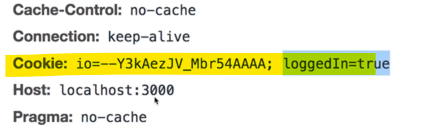
Lý do tại sao lại có Session & Cookie

Chính vì giao thức HTTP là giao thức stateless, nghĩa là khi request sent & response returned thì đều là 1 trang giấy trắng

Client & Server không hay biết ai đã gửi bla bla



Bây giờ để gửi thông tin đi kèm với request 🡪 ta sử dụng header

Và trong đó bao gồm cả Cookie: chính là thông tin được gửi từ Client đến Server

Cookie và Session được ứng dụng nhiều nhất cho hệ thống login

Nếu chỉ sử dụng HTTP để đọc thông tin đã authenticated 🡪 thì mỗi lần gửi request đến server đã lại phải đăng nhập lại mỗi lần gửi

Thế bây giờ để tiện thì ta lưu thông tin vào trong Cookie luôn để không phải nhập đi nhập lại đúng không !!

* Hoàn toàn chính xác NHƯNG

NHƯNG nên nhớ rằng Cookie được set ở Client. Điều này có nghĩa là nếu ta chứa User Credential (username & password) 🡪 vào cookie thì khả năng bị đánh cắp là 100%

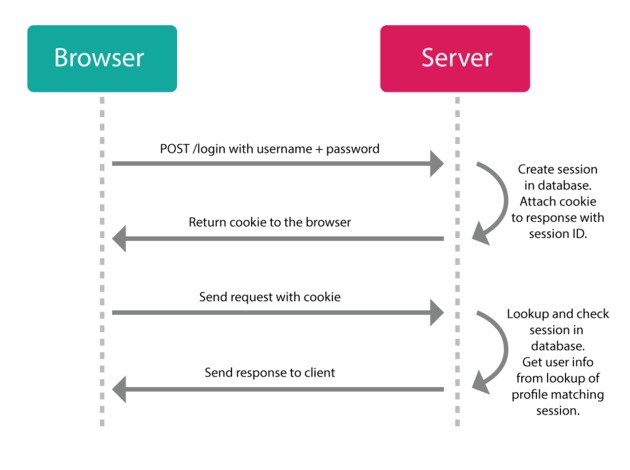
Vậy bây giờ ta chỉ gửi 1 thông tin rằng isLogin = true/false để xác nhận thì sao 🡪 hoàn toàn khả thi nhưng Client có thể thay đổi trạng thái (cookie) thì việc Authentication này chả có ý nghĩa cm gì ☹

* Chính vì thế Session ra đời

Session

Là một thông tin dưới dạng tệp được lưu ở Server, chỉ được tạo & modify ở Server

Ta dùng Session như cái gì đó đối chiếu với giá trị ở Cookie



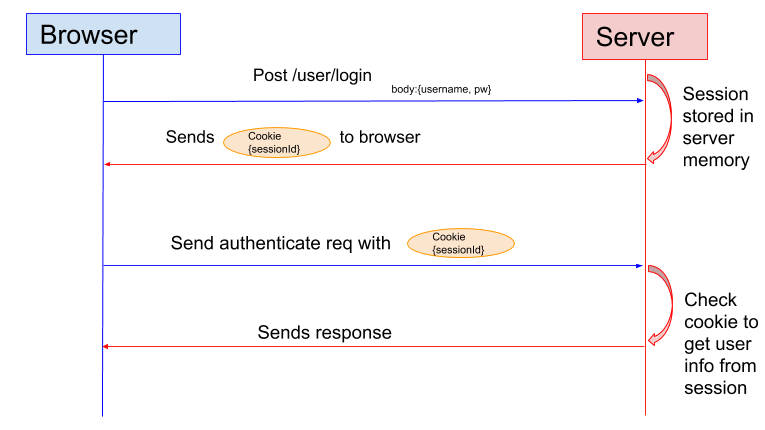
Tham khảo hình dưới đây để hiểu thêm

# So sánh giữa Cookie và Session

| **Cookie** | **Session session là gì session là gì** |
| --- | --- |
| **Cookie** được lưu trữ trên trình duyệt của người dùng. | **Session** không được lưu trữ trên trình duyệt. |
| Dữ liệu **cookie** được lưu trữ ở phía client. | Dữ liệu **session** được lưu trữ ở phía server. |
| Dữ liệu **cookie** dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client. | Dữ liệu **session** không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. |
| Dữ liệu **cookie** có sẵn trong trình duyệt đến khi expired. | Sau khi đóng trình duyệt sẽ hết phiên làm việc (session) |

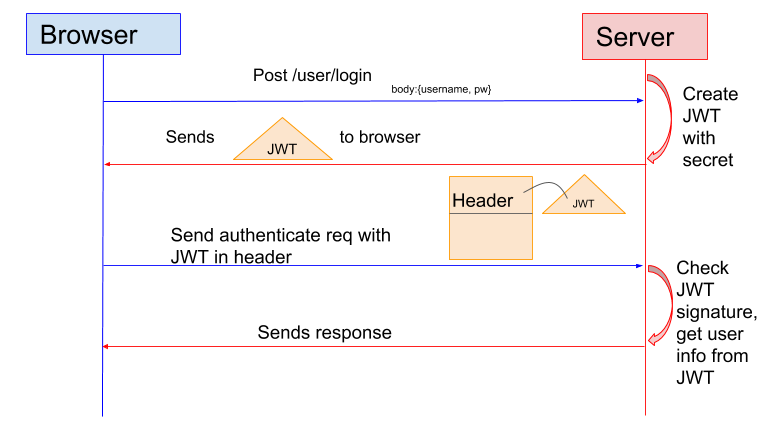
## Session Based Authentication

Trong ***Session Based Authentication***, server sẽ tạo một session cho người dùng sau khi người dùng đăng nhập. Session ID sẽ được lưu ở cookie trong trình duyệt của người dùng. Trong khi người dùng vẫn còn đăng nhập, cookie sẽ gửi tiếp cùng với những request tiếp theo. Server có thể so sánh session ID lưu trữ ở cookie với session được lưu trong bộ nhớ để xác minh thông tin người dùng và gửi phản hồi với trạng thái tương ứng



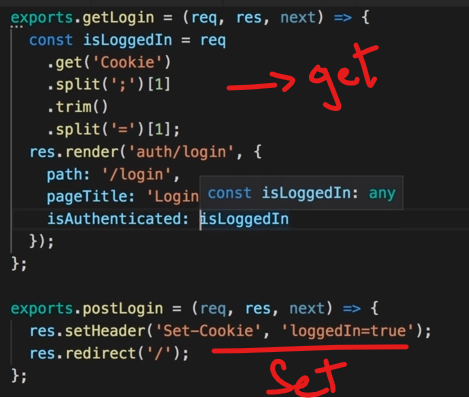
## Token Based Authentication

Một số trang website sử dụng JSON Web Token (JWT) thay thế cho việc xác thực bằng session. Trong ***Token Based Authentication***, server tạo JWT một cách bí mật và gửi JWT tới client. Client lưu JWT (thường là local storage) và thêm JWT vào header với mọi request. Server sau đó sẽ xác thự JWT với mọi request từ client và trả về response.

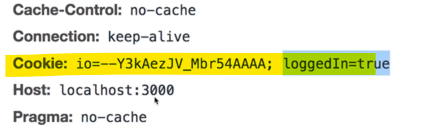


# Vanilla Cookie

Không dùng express-session

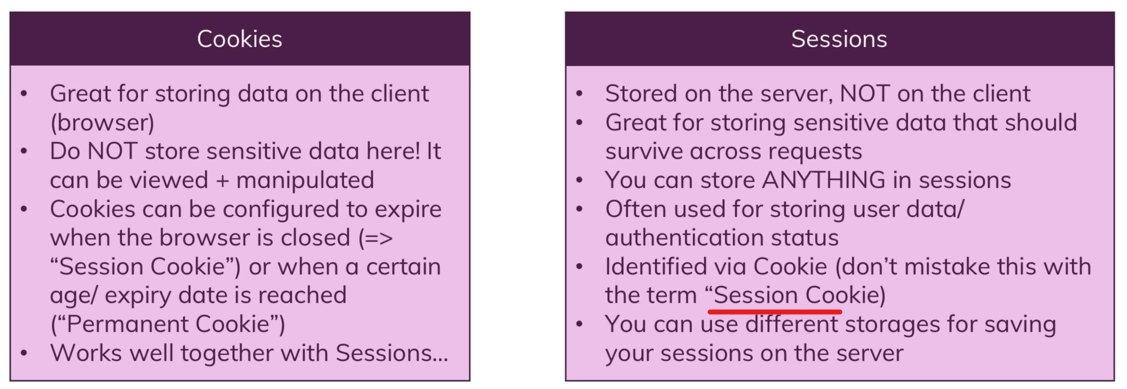


Sau khi đăng nhập thành công, session ở phía server sẽ tạo set of cookies gọi là Session Cookie. ^~ sessionCookie có 1 unique ID



In browser :

Và cuối cùng là



Session Cookie là cookie sẽ hủy ngay khi browser tắt