



FPT POLYTECHNIC



www.poly.edu.vn



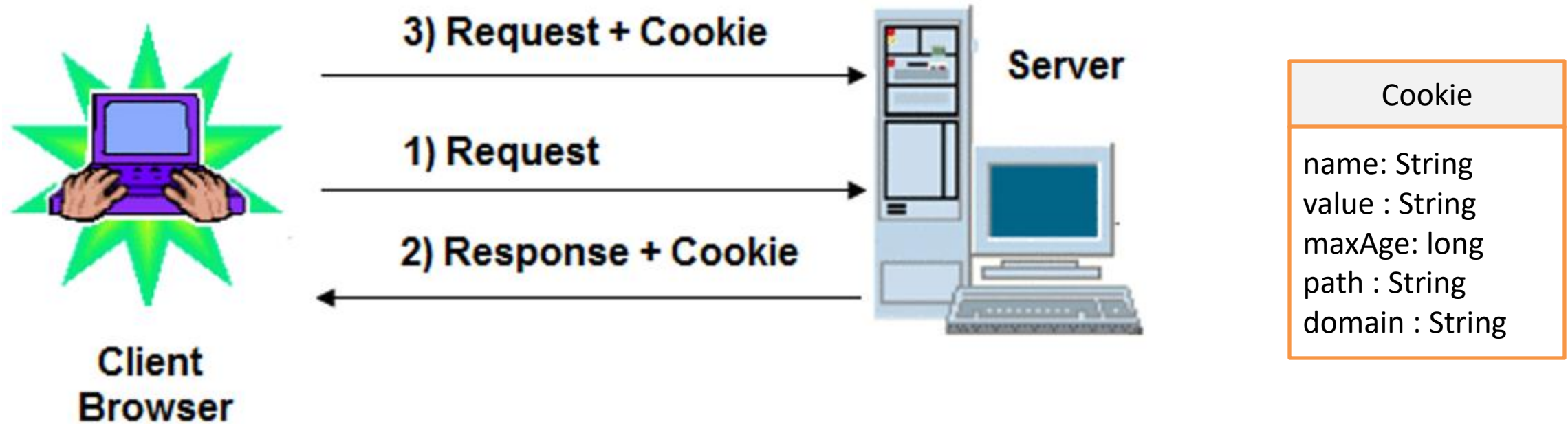
Conceive Design Implement Operate

BEANUTIL, EMAIL, SESSIONS VÀ COOKIES

LẬP TRÌNH JAVA #3 (P5.2)

- ☐ Giới thiệu Cookie
- ☐ Tạo và gửi Cookie về Client
- ☐ Đọc Cookie
- ☐ Sử dụng Base64 để mã hóa và giải mã giá trị cookie
- ☐ Session và Scopes



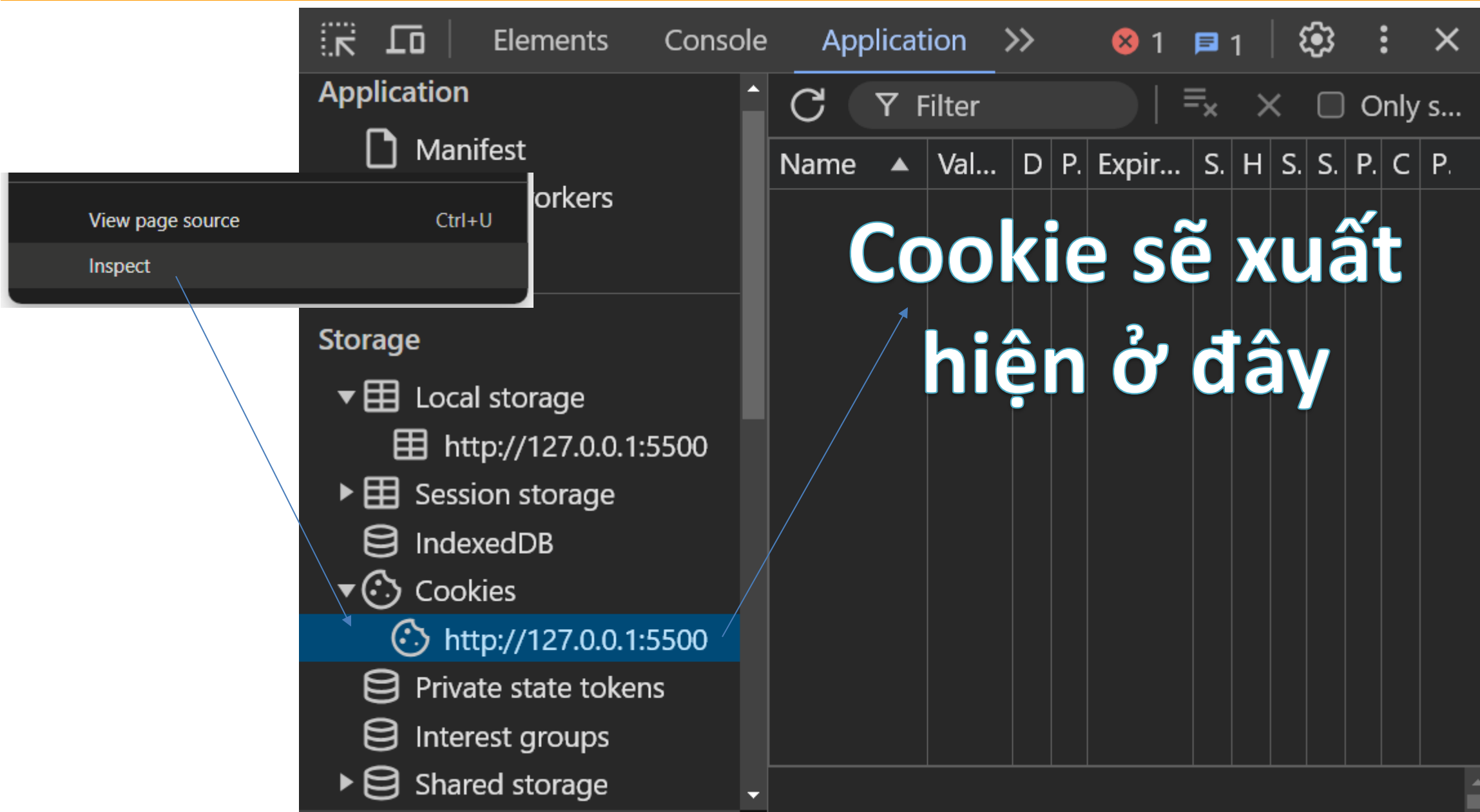


- ❑ Cookie là mẫu tin văn bản nhỏ được lưu trữ bởi trình duyệt và truyền thông với web server thông qua request và response.
- ❑ Các thuộc tính của một cookie gồm name, value, max age, path, domain...
- ❑ Kích thước mỗi cookie không vượt quá 4KB

```
Cookie cookie = new Cookie("ck-name", "ck-value"); // tạo cookie
cookie.setMaxAge(10*60*60); // thời hạn tồn tại là 10h
cookie.setPath("/"); // có hiệu lực toàn site
resp.addCookie(cookie); // gửi về client
```

❑ Chú ý:

- ❖ MaxAge: Tuổi thọ (tính bằng giây) tức là thời hạn có hiệu lực của cookie. Hết thời hạn này thì cookie sẽ bị trình duyệt hủy bỏ.
- ❖ Path: Phạm vi đường dẫn cookie được phép truyền lên web server.
- ❖ Gọi phương thức addCookie của HttpServletResponse để gửi cookie về trình duyệt.



The screenshot shows the Chrome DevTools Application tab. The 'Storage' section is expanded, and the 'Cookies' folder is selected. A specific cookie for the URL 'http://127.0.0.1:5500' is highlighted. A context menu is open over the 'Inspect' button, and a large text overlay reads 'Cookie sẽ xuất hiện ở đây' (Cookie will appear here).

Application

Manifest

View page source Ctrl+U

Inspect

Storage

- Local storage
 - http://127.0.0.1:5500
- Session storage
- IndexedDB
- Cookies
 - http://127.0.0.1:5500
- Private state tokens
- Interest groups
- Shared storage

Cookie sẽ xuất hiện ở đây

Name	Val...	D	P.	Expir...	S.	H	S.	S.	P.	C	P.
------	--------	---	----	----------	----	---	----	----	----	---	----

```
Cookie[] cookies = req.getCookies();  
if(cookies != null) {  
    for(Cookie cookie: cookies) {  
        String name = cookie.getName();  
        String value = cookie.getValue();  
    }  
}
```

- ❑ Cookie được gửi kèm trong mỗi request. Sử dụng phương thức `getCookies()` để lấy mảng chứa tất cả các cookie trong request. Sau đó duyệt và xử lý từng cookie.

- ❑ Nếu value của cookie chứa tự đặc biệt hoặc utf-8 thì cần mã hóa dạng Base64 trước khi gửi về web browser.

```
String text = "Thực học - Thực Nghiệp";  
byte[] bytes = text.getBytes(); // text => byte[]  
String encodedText = Base64.getEncoder().encodeToString(bytes);  
// encodedText là VGjhu7FjIGjhu41jIC0gVGjhu7FjIE5naGnhu4dw
```

- ❑ Sau khi đọc giá trị của cookie bạn cần phải giải mã

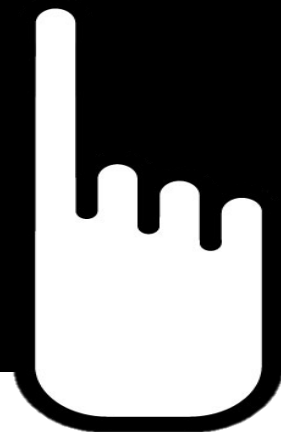
```
String encodedText = "VGjhu7FjIGjhu41jIC0gVGjhu7FjIE5naGnhu4dw";  
byte[] bytes = Base64.getDecoder().decode(encodedText);  
String decodedText = new String(bytes);  
System.out.println(decodedText);  
// decodedText là "Thực học - Thực Nghiệp"
```

- ☐ Ghi nhớ tài khoản đăng nhập để lần sau không phải nhập lại
- ☐ Ghi nhớ các liên kết (URL) đã xem (ví dụ các bản tin đã xem)
- ☐ Ghi nhớ các mặt hàng đã chọn trong giỏ hàng
- ☐ Và tất cả những gì bạn cần ghi nhớ trên từng máy khách



DEMO

Lập trình Cookie
Ghi nhớ tài khoản



- ❑ HttpSession (còn gọi là session) là đối tượng được sử dụng để duy trì thông tin tích lũy từ nhiều request của mỗi phiên làm việc.
- ❑ Session được tạo ra và quản lý trên web server. Nó sẽ được giải phóng khỏi bộ nhớ sau một khoảng thời gian (mặc định là 20 phút) người dung không có tương tác nào hoặc đóng trình duyệt.
- ❑ Tham chiếu đến session trong servlet
 - ❖ `HttpServletRequest: request.getSession()`
 - ❖ `HttpServlet: this.getSession()`
- ❑ Tham chiếu đến session trong servlet
 - ❖ `${pageContext.request.session}`
 - ❖ Hoặc `${pageContext.session}`

Method	Description
getId(): Long	Lấy mã của session. Mỗi session được tạo ra có một id duy nhất.
getCreateTime(): Date	Lấy thời điểm tạo session
getLastAccessedTime(): Date	Lấy thời điểm truy xuất website cuối cùng
getMaxInactiveInterval(): long	Lấy thời hạn (timeout), tính theo mili giây
setMaxInactiveInterval(long)	Thay đổi timeout
getAttribute(String): Object	Lấy attribute trong session
setAttribute(String, Object)	Thêm attribute vào session
removeAttribute()	Xóa attribute khỏi session
Invalidate()	Xóa tất cả attribute khỏi session

- ❑ Thêm attribute vào session
 - ❖ `session.setAttribute(name, value)`
- ❑ Xóa attribute khỏi session
 - ❖ `session.removeAttribute(name)`
- ❑ Đọc giá trị attribute từ session
 - ❖ `session.getAttribute(name): Object`

❑ Thêm attribute vào session

❖ `<c:set var="name" value="${value}" scope="session"/>`

❑ Xóa attribute khỏi session

❖ `<c:remove var="name" scope="session"/>`

❑ Đọc giá trị attribute từ session

❖ `${sessionScope.name}`

- ❑ Dựa trên yêu tố kỹ thuật là duy trì và chia sẻ dữ liệu theo phiên, lập trình viên có thể ứng dụng vào các công việc sau:
 - ❖ Duy trì giỏ hàng để chia sẻ cho nhiều servlet/jsp trong phiên
 - ❖ Duy trì thông tin đăng nhập
 - ❖ Cache dữ liệu cá nhân theo phiên



- ✓ Giới thiệu Cookie
- ✓ Tạo và gửi Cookie về Client
- ✓ Đọc Cookie
- ✓ Sử dụng Base64 để mã hóa và giải mã giá trị cookie
- ✓ Session và Scopes





Cảm ơn