

Môn học: Lập trình Web

LAB 04

Cookie, Session, Web Services

I. Mục tiêu

- Hiểu biết về Cookie và Session, các khái niệm cơ bản, lợi ích và cách thức hoạt động.
- Ứng dụng Cookie và Session để hiện thực mô hình xác thực người dùng cơ bản cho ứng dụng Web.
- Hiểu biết về Web Services và các loại Web Services phổ biến: REST, SOAP.
- Hiểu biết các mã trạng thái của giao thức HTTP (HTTP Status Codes) và xử lý các mã trạng thái khi làm việc với Web service.
- Hiểu biết về các chuẩn định dạng dữ liệu thường gặp: JSON (JavaScript Object Notation) và XML (eXtensible Markup Language).

II. Các bước thực hiện

Chú ý: các file liên quan đến bài Lab và tài liệu tham khảo được đặt trong thư mục “refs”

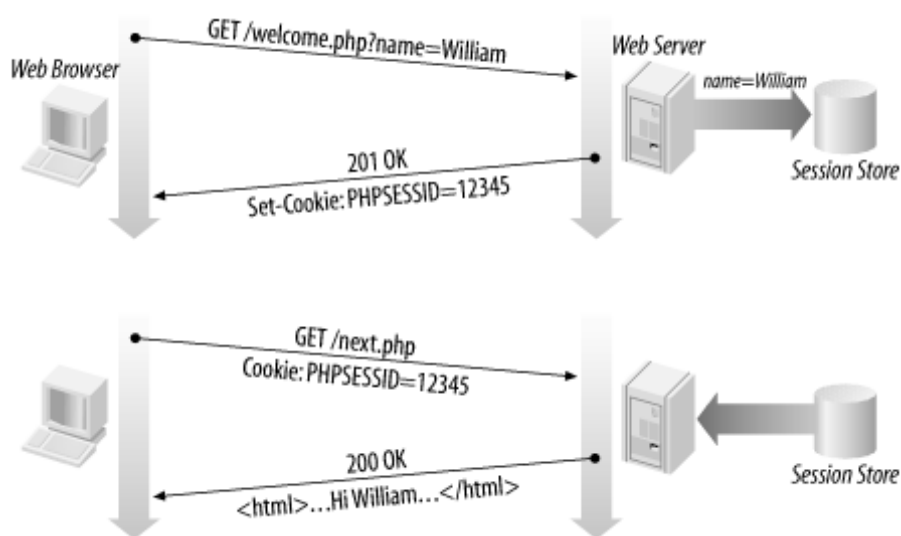
A. Phần 1

1. Tìm hiểu Cookie

- Cookie là một phần thông tin nhỏ được gửi từ server đến browser trong lần request đầu tiên, lưu trữ ở browser và được gửi đến server trong những lần request tiếp theo.
- Cookie được sử dụng trong các tính năng:
 - Xác thực người dùng.
 - Lưu giữ các cấu hình của người dùng, ứng dụng giỏ hàng,...

2. Tìm hiểu Session

- Session là được hiểu là 1 phiên làm việc của người sử dụng giao tiếp với 1 ứng dụng, bao gồm một chuỗi HTTP request/response giữa một Web browser và server



3. Cách thức truy xuất Cookie và Session

○ Cookie

Để truy xuất danh sách các Cookie được lưu trên browser của trang Web hiện tại, ta sử dụng cú pháp như sau:

➤ **Ngôn ngữ javascript:**

```
var cookies = document.cookie;
```

➤ **Ngôn ngữ PHP:**

```
$cookies = $_COOKIE;
```

Xem thêm các hàm/phương thức hỗ trợ truy xuất (đọc, ghi) Cookie ở địa chỉ:

- **javascript:** http://www.w3schools.com/js/js_cookies.asp
- **PHP:** http://www.w3schools.com/php/php_cookies.asp

○ Session

Để truy xuất Session trong PHP, ta có thể dùng biến `$_SESSION`

```
$session = $_SESSION
```

Xem thêm các hàm hỗ trợ truy xuất Session trong ngôn ngữ PHP ở địa chỉ:

http://www.w3schools.com/php/php_sessions.asp

Function	Description
<code>session_destroy()</code>	removes all data associated with current session
<code>session_id()</code>	returns session ID number for current session
<code>session_regenerate_id()</code> <code>session_regenerate_id(boolean)</code>	replace current session ID with a new one; if TRUE is passed, also wipes any data associated with the old one
<code>session_save_path()</code> <code>session_save_path("directory")</code>	gets/sets the folder name on the server where session data is stored (can also be configured in PHP.ini)
<code>session_start()</code>	begins a new session for the current client; if a session is already in progress, has no effect
<code>session_status()</code>	returns information about whether session support is enabled or disabled on the current server
<code>session_unset()</code>	frees all session variables currently registered

4. Các loại lưu trữ dữ liệu khác trên trình duyệt web

- Local Storage – [Tham khảo](#)
- Session Storage – [Tham khảo](#)
- IndexedDB – [Tham khảo](#)
- WebSQL – [Tham khảo](#)

❖ Bài tập phần 1

Bài 1. (thư mục phan1_bai1) Nêu các ưu điểm, hạn chế của Cookie và Session, liệt kê các ví dụ có sử dụng Cookie và Session trong ứng dụng Web thực tế, trả lời vào file Word.

Bài 2. (thư mục phan1_bai2) Hiện thực các tính năng cho phép quản lý cookie của website (hiển thị danh sách cookie hiện tại, thêm sửa, xóa thông tin cookie).

- **Gợi ý:** Xem và chỉnh sửa cookie thông qua đối tượng document.cookie bằng mã javascript
- **Tham khảo:** Thêm cookie với javascript

```
document.cookie = "teste=5; domain=.cse.ust.hk";  
document.cookie = "testf=6; path=/comp4632/lab9_test";  
document.cookie = "testg=7; domain=.cse.ust.hk; path=/comp4632/lab9_test";  
document.cookie = "testh=8; Expires= Fri, 01 Jan 2016 00:00:00 GMT";  
document.cookie = "testi=9; HttpOnly";  
document.cookie = "testj=10; Secure";
```

Bài 3. (thư mục phan1_bai3) Hiện thực các tính năng như bài 2 nhưng sử dụng loại lưu trữ **Local Storage** và **IndexedDB** (tự thiết kế cấu trúc bảng dữ liệu).

Bài 4. (thư mục phan1_bai4) Ứng dụng PHP Session & Cookie để hiện thực một hệ thống xác thực người dùng đơn giản bằng username và password. Hệ thống có hỗ trợ tính năng **ghi nhớ đăng nhập (remember me)**, và giải thích cách thức hoạt động của tính năng này (trong file word).

❖ **Yêu cầu:** SV cần hiện thực 3 files:

- **login.php**
 - Nếu người dùng chưa đăng nhập, thì hiển thị một form đăng nhập với 2 trường username và password, và checkbox **ghi nhớ đăng nhập**.
 - Nếu người dùng đã đăng nhập thành công, chuyển hướng trình duyệt đến trang *info.php*
 - Tính năng **ghi nhớ đăng nhập** (tham khảo [tại đây](#))
- **info.php**
 - Nếu người dùng chưa đăng nhập, chuyển hướng trình duyệt đến trang *login.php*
 - Nếu người dùng đã đăng nhập thành công trước đó, hiển thị thông tin là tên đăng nhập của người dùng.
- **logout.php**
 - Tính năng đăng xuất người dùng khỏi hệ thống

B. Phần 2

1. Tìm hiểu Web Services

- Xem lại phần lý thuyết Web Services
- Các phương thức của giao thức HTTP, tham khảo: **HTTP_Request_Methods.pdf**
- Danh sách mã trạng thái của giao thức HTTP:, tham khảo: **HTTP_Status_Codes.pdf**

2. Sử dụng Postman để kiểm thử web service

- Postman là một phần mở rộng (extension), cho phép làm việc với các web service (như REST), hỗ trợ cho việc kiểm thử. Chương trình hỗ trợ tất cả các phương thức của HTTP (GET, POST, PUT, DELETE, OPTIONS, HEAD,...) Postman cũng hỗ trợ lưu lại các lần sử dụng.
- Các bước sử dụng:

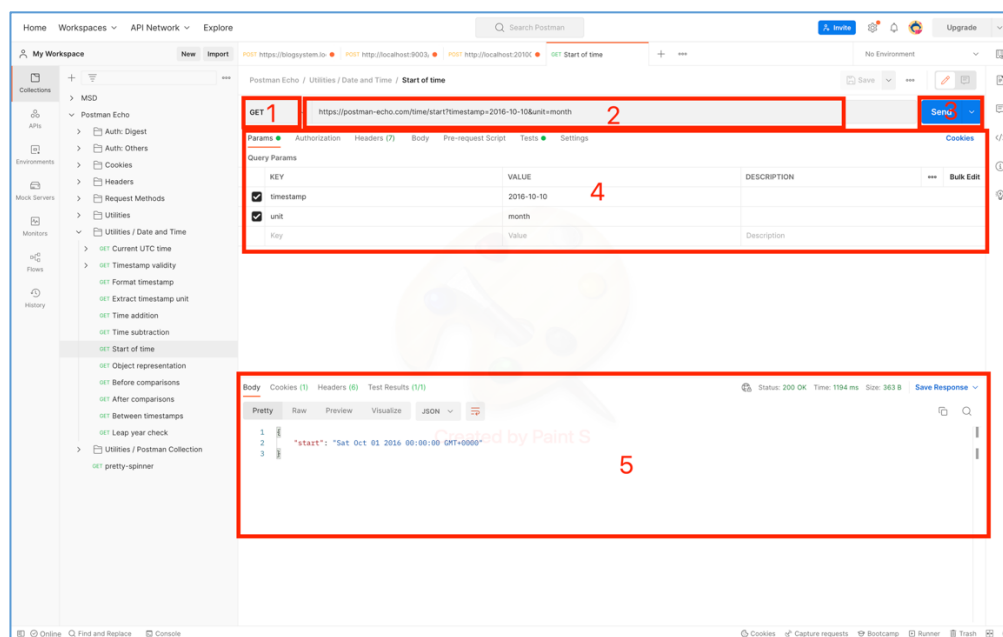
Bước 1. Thiết lập phương thức cho request HTTP

Bước 2. Nhập End Point

Bước 3. Nhập các tham số tương ứng cho request

Bước 4. Nhấn nút Send để bắt đầu thực hiện request đến web service

Bước 5. Kiểm tra kết quả (status code, response headers, body,...)



3. Các hàm PHP cơ bản

- **json_encode:** <https://www.php.net/manual/en/function.json-encode.php>
- **json_decode:** <https://www.php.net/manual/en/function.json-decode.php>
- **header:** <https://www.php.net/manual/en/function.header.php>
- **http_response_code:** <https://www.php.net/manual/en/function.http-response-code.php>
- **headers_list:** <https://www.php.net/manual/en/function.headers-list.php>

❖ Bài tập phần 2

Bài 1. (phan2_bai1) Trình bày định nghĩa, ưu điểm và nhược điểm của các loại Web Service: REST và SOAP, trả lời vào file Word.

Bài 2. (phan2_bai2) Hiện thực Web Services (RESTful + JSON response data type), sau đó sử dụng AJAX cùng với các dịch vụ đã xây dựng để hiện thực trang web có các tính năng sau đây (Thực hành bài Lab với cơ sở dữ liệu được cho trong bài Lab 03), kiểm thử các web services đã xây dựng bằng Postman:

- a. (a.php) Tính năng liệt kê tất cả các dữ liệu có trong bảng **products** (hiển thị trong một bảng, HTML table).
- b. (b.php) Tính năng thêm một record dữ liệu mới vào bảng **products** (dữ liệu được người dùng nhập vào từ form, các dữ liệu phải được kiểm tra theo đúng yêu cầu bằng ngôn ngữ PHP).
- c. (c.php) Tính năng chỉnh sửa một record dữ liệu trong bảng **products**.
- d. (d.php) Tính năng xóa một record dữ liệu khỏi bảng **products**.

Ghi chú kiểm tra dữ liệu đầu vào:

- **id** : số nguyên.
- **name** : kiểu chuỗi, độ dài từ 5 đến 40 kí tự.
- **description** : kiểu chuỗi, độ dài tối đa 5000 kí tự.
- **price**: kiểu số thực
- **image**: kiểu chuỗi, độ dài tối đa 255 kí tự.

III. Cách thức nộp bài

- Bài Lab 04 bao gồm 2 phần:
 - **Phần làm và nộp tại lớp (bài 1,2,3 của phần 1)**: sinh viên làm và nộp tại lớp tại mục “Bài nộp tại lớp (Lab 02)”, tạo thư mục MSSV chứa các thư mục bài làm (phan1_bai1, phan1_bai2, phan1_bai3), sau đó nén thư mục trên vào một file **zip** (<mssv>.zip) và nộp.
 - **Phần bài nộp sau**: sinh viên tạo thư mục MSSV chứa các thư mục bài làm của các phần còn lại, sau đó nén thư mục trên vào một file **zip** (<mssv>.zip) và nộp ở LMS: <https://lms.hcmut.edu.vn/> trong trang của môn học tương ứng.
- Các bài nộp sai quy định sẽ không được tính điểm.
- Các bài làm giống nhau sẽ bị xem là gian lận và bị 0 điểm,
- Sinh viên theo dõi deadline nộp bài lab và nộp bài đúng hạn để được tính điểm, mọi trường hợp nộp bài trễ hạn sẽ không được giải quyết.
- **Chỉ nhận bài nộp thông qua BKel, không nhận bài nộp qua email hay các hình thức khác.**

--HẾT--