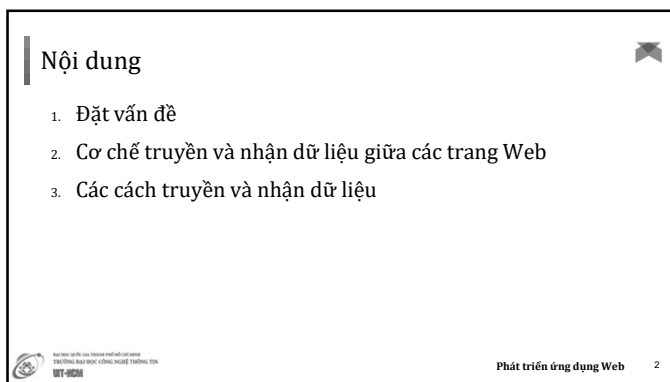


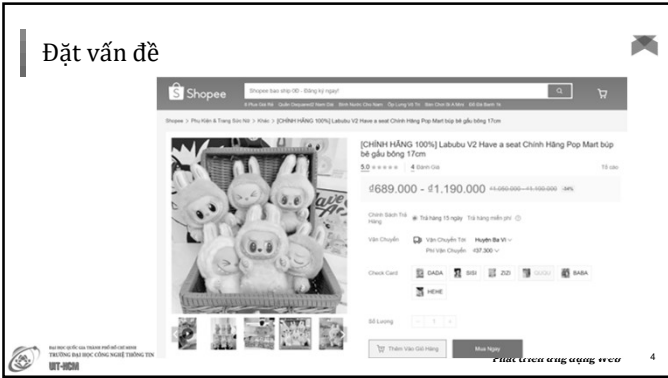
1



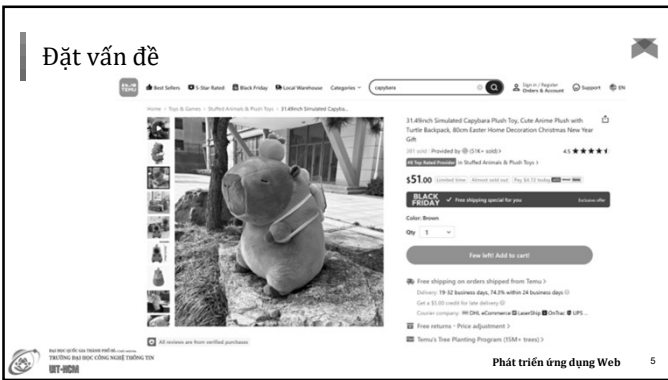
2



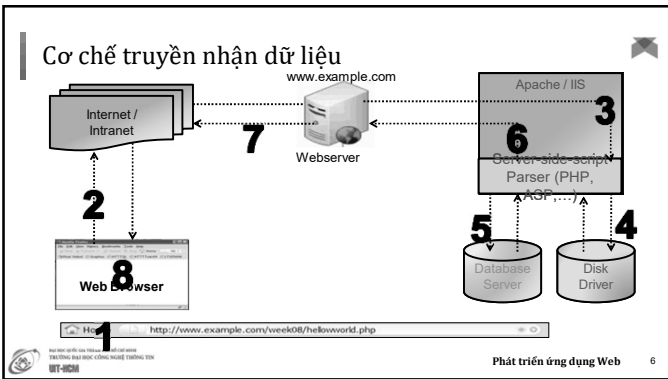
3



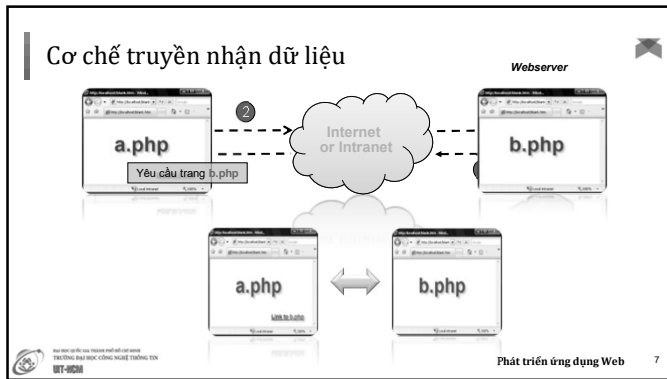
4



5



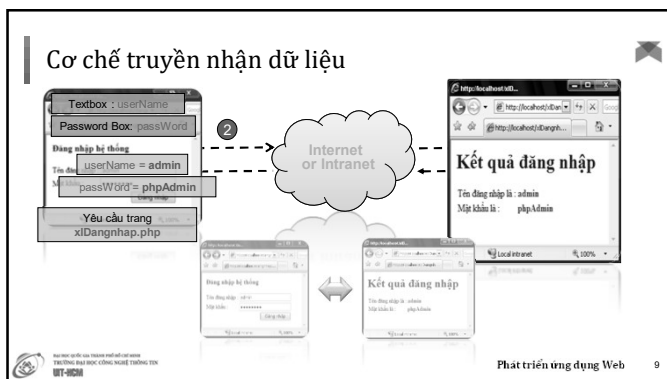
6



7



8



9

Cơ chế truyền nhận dữ liệu

- Dữ liệu của người dùng từ trình duyệt sẽ được gửi lên máy chủ dưới dạng từng cặp biến=giá trị và có thể đi theo 4 cách khác nhau
- Tùy theo từng cách gửi lên máy chủ mà máy chủ có các cách tương ứng để lấy dữ liệu được gửi lên
- 4 cách truyền dữ liệu:
 - GET, POST và
 - COOKIES, Session.
- Vậy GET, POST và COOKIES, Session là gì?

10

Truyền nhận dữ liệu

- GET
- POST
- **Trình duyệt web** giao tiếp với server bằng cách sử dụng một trong hai phương thức **HTTP (Hypertext Transfer Protocol)** - GET và POST.

11

Truyền nhận dữ liệu


- Trang web **nhập** dữ liệu:
 - Sử dụng đối tượng <form>
 - Nhập dữ liệu thông qua các <control>
 - Thực hiện việc truyền dữ liệu thông qua nút nhấn Submit

```
<form name="form1" method="post/get" action="URL" >
  <input type="submit" name="Submit" value="Submit" >
</form>
```

12

Truyền nhận dữ liệu

- Trang **nhận** dữ liệu (URL) sử dụng các biến toàn cục của PHP
 - `$_POST["tên control"]`
 - `$_GET["tên control"]`
 - `$_REQUEST["tên control"]`;


 Học viện quốc gia thông tin và công nghệ
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
UIT-HCM

Phát triển ứng dụng Web 13

13

Truyền nhận dữ liệu bằng GET

- Tham số truyền đi qua địa chỉ URL
`http://domain/pathfile?fieldname1=value1&fieldname2=value2`
- Ví dụ:
`http://www.uit.edu.vn/xemdiem.php?mssv=11520123`


 Học viện quốc gia thông tin và công nghệ
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
UIT-HCM

Phát triển ứng dụng Web 14

14

Truyền nhận dữ liệu bằng GET

- Truyền thông tin qua form
 - Form có thuộc tính `method="GET"`
 - Nhận dữ liệu thông qua mảng toàn cục:
 - `$_GET["tên control"]`
 - `$_REQUEST["tên control"]`

 Học viện quốc gia thông tin và công nghệ
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
UIT-HCM

Phát triển ứng dụng Web 15

15

Ví dụ: Trang nhập dữ liệu

localhost/get_post_session_cookie/get/login.php

User name

Pass word

Login

16

Ví dụ: Trang nhận dữ liệu

localhost/get_post_session_cookie/get/index.php?USER=admin&PASS=1234567&Login=Login

user name:admin

Pass word:1234567

17

Trang nhập dữ liệu (login.php)

```
<form name="form1" method="get" action="index.php">
  <p>
    <label>User name
    <input name="USER" type="text" id="USER">
  </label>
  </p>
  <p>
    <label>Pass word
    <input name="PASS" type="text" id="PASS">
  </label>
  </p>
  <p>
    <input type="submit" name="Submit" value="Login">
  </p>
</form>
```

18

Trang nhận dữ liệu (index.php)

```
<?php
    echo "user name:".$_GET["USER"]."</br>";
    echo "Pass word:".$_GET["PASS"]."</br>";
?>

echo "User name: " . htmlspecialchars($_GET["USER"], ENT_QUOTES, 'UTF-8');
```

19

Truyền nhận dữ liệu bằng GET

- Truyền dữ liệu bằng phương thức GET không thông qua form mà truyền trực tiếp thông qua địa chỉ URL
- Ví dụ: <http://localhost/chitietsanpham.php?msp=001>

20

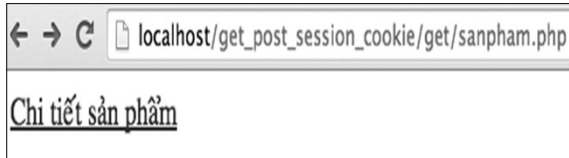
Truyền nhận dữ liệu bằng GET

- Trong **phương thức GET**, dữ liệu được gửi dưới dạng các **tham số URL**
- Có thể đưa lên nhiều cặp tên `biến=giá_trị` bằng cách phân cách chúng bởi dấu `&`: <http://localhost/chitietsanpham.php?masp=7&manx=12>
- Với địa chỉ URL trên, gửi lên 2 cặp `biến=giá_trị` theo phương thức GET: `masp=7, manx=12`
- Lấy giá trị thông qua mảng toàn cục:
 - `$_GET["tên biến"]`
 - `$_REQUEST["tên biến"]`

21

Ví dụ

- Xét trang "sanpham.php" có giao diện sau:



22

Ví dụ

- Khi click chọn link "chi tiết sản phẩm" thì trang "chitietsanpham.php" được mở lên. Trang "chitietsanpham.php" lấy dữ liệu từ trang "sanpham.php" và hiển thị lên màn hình



23

Ví dụ

- Trang "sanpham.php"


```
<head>
  <meta charset="UTF-8">
  <title>Sân phẩm</title>
</head>
<body>
  <a href="chitietsanpham.php?masp=12">Chi tiết sản phẩm </a>
</body>
</html>
```

24

Ví dụ

- Trang lấy dữ liệu "chitietsanpham.php"

```
<html>
<head>
<meta charset="UTF-8">
<title>Thông tin chi tiết sản phẩm</title>
</head>
<body>
<?php
    $masp = $_REQUEST["masp"];
    echo "Mã sản phẩm lấy được:". $masp;
?>
</body>
</html>
```



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN


UT-HCM

Phát triển ứng dụng Web25

25

Truyền nhận dữ liệu bằng POST

- Form có thuộc tính method = "POST"
- Tham số truyền đi được ẩn bên trong FORM
- Lấy dữ liệu:
 - \$_POST["tên control"];
 - \$_REQUEST["tên control"];



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

UT-HCM

Phát triển ứng dụng Web26

26

Ví dụ

- Xét trang đăng ký tài khoản có giao diện sau:


Tài khoản đăng nhập

Mật khẩu đăng nhập

Nhập lại mật khẩu

Giới tính

Đăng ký



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

UT-HCM

Phát triển ứng dụng Web27

27

Ví dụ

- Khi nhập chọn dữ liệu trên form đăng ký và nhấn chọn nút đăng ký thì website mở trang và hiển thị thông tin đã nhập trong trang "thongtindangky.php"

Phát triển ứng dụng Web 28

28

Ví dụ

- Nội dung được hiển thị trong trang "thongtindangky.php"

Phát triển ứng dụng Web 29

29

Ví dụ - trang "dangky.php"

```
<table align="center" bgcolor="#F0F0F0" cellpadding="0" cellspacing="0" border="0" width="250"
  cellpadding="15">
<tr><td>
<form method="POST" action="thongtindangky.php">
  <input type="text" placeholder="Tài khoản đăng nhập" name="user" size="40"><br>
  <input placeholder="Mật khẩu đăng nhập" type="password" name="pass" size="40"><br>
  <input placeholder="Nhập lại mật khẩu" type="password" name="repass" size="40"><br>
  <select name="gt">
    <option selected>Giới tính </option>
    <option value="Nam"> Nam</option>
    <option value="Nu"> Nữ</option>
  </select><br>
  <input type="Submit" name="Submit" value="Đăng ký">
</form>
</td></tr>
</table>
```

Phát triển ứng dụng Web 30

30

Ví dụ - trang "thongtindangky.php"

```
<body>
<?php
    $user= $_POST["user"];
    $pass= $_POST["pass"];
    $gt=$_POST["gt"];
    echo "<h3>Bạn đã đăng ký với tài khoản sau:</h3>";
    echo "Tên đăng nhập:". $user."<br>";
    echo "Mật khẩu đăng nhập:". $pass."<br>";
    echo "Giới tính:". $gt."<br>";
?>
</body>
```

31

Truyền/Nhận dữ liệu từ Checkbox

File: Checkbox.php

```
<html><body>
<form method="get" action="checkbox.php">
    <input type="checkbox" name="chk1" value="en">English <br>
    <input type="checkbox" name="chk2">Vietnam<br>
    <input type="submit" value="submit"><br>
</form>
<?php
    echo "checkboxbox 1 : " . $_REQUEST['chk1'];
    echo "checkboxbox 2 : " . $_REQUEST['chk2'];
?>
</body></html>
```



32

Truyền/Nhận dữ liệu từ Radio Button

File: RADIO.PHP

```
<html><body>
<form action="radio.php" method="GET">
    <input type="RADIO" NAME="radGT">Nam<br>
    <input type="RADIO" NAME="radGT" VALUE="Nu">Nữ<br>
    <input type="SUBMIT" VALUE="Submit">
</form>
<?php
    if (isset($_GET['radGT'])) {
        echo "Giới tính : " . $_GET['radGT'];
    }
?>
</body></html>
```



33

Truyền/Nhận dữ liệu từ ComboBox

```
<body>
<form method="POST" action="select.php">
  <select name="noicongtac">
    <option value="TPHCM">Thành Phố Hồ Chí Minh</option>
    <option value="HN">Hà nội</option>
    <option>Đà Nẵng</option>
  </select>
  <input type="submit" name="submit" value="Chọn"/>
</form>
<?php
if (isset($_POST['noicongtac'])) {
    echo "Bạn đã chọn: " . $_POST['noicongtac'] . "<br/>";
}
?>
</body>
```

34

Truyền/Nhận dữ liệu từ Listbox – dùng mảng

File: listbox.PHP

```
<body>
<form method="POST" action="listbox.php">
  <select name="noicongtac[]" multiple>
    <option value="TPHCM">Thành Phố Hồ Chí Minh</option>
    <option value="HN">Hà nội</option>
    <option>Đà Nẵng</option>
  </select>
  <input type="submit" name="submit" value="Chọn"/>
</form>
<?php
if (isset($_POST['submit'])) {
    if ($_POST['submit'] == "Chọn") {
        echo "Bạn đã chọn:<br>";
        foreach($_POST['noicongtac'] as $nct)
            echo $nct."<br>";
    }
}
?>
</body>
```

35

An toàn dữ liệu

- Dùng hàm htmlspecialchars(), filter_input() hoặc htmlentities() để mã hóa đầu ra
 - Dùng htmlspecialchars() để chuyển đổi ký tự đặc biệt cơ bản. Ví dụ: echo "Hello, " . htmlspecialchars(\$_GET['name'], ENT_QUOTES, 'UTF-8');
- (thay vì: echo "Hello, " . \$_GET['name'];)
- filter_input(): Lấy và lọc dữ liệu trực tiếp từ các nguồn đầu vào (GET, POST, COOKIE, SERVER). Có thể kết hợp Lọc dữ liệu khi nhận vào (filter_input) và dùng htmlspecialchars để Sanitize dữ liệu trước khi hiển thị.
 - htmlentities(): Chuyển đổi tất cả các ký tự có thể thành HTML entities. Chậm hơn htmlspecialchars vì chuyển đổi nhiều ký tự hơn.

36

An toàn dữ liệu

- Ví dụ: Dùng filter_input() để lọc dữ liệu đầu vào từ \$_POST\

```
<?php
$username = filter_input(INPUT_POST, 'username', FILTER_SANITIZE_STRING);
$password = filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING);
echo "Username: " . htmlspecialchars($username, ENT_QUOTES, 'UTF-8');
?>
```

37

Khi nào dùng GET

- Ưu điểm:
 - Dữ liệu được truyền đi nhanh hơn POST
 - Có thể giả lập phương thức GET để truyền dữ liệu mà không cần dùng FORM. Ví dụ: <http://example.com/search.php?query=laptop>
 - Có thể đánh dấu trang (bookmark) bằng các giá trị chuỗi truy vấn cụ thể.
- Nhược điểm:
 - Không thích hợp để truyền dữ liệu có tính bảo mật
 - Dữ liệu truyền đi bị giới hạn (2048 ký tự - 8KB tùy trình duyệt)
- Dùng cho các thao tác truy vấn, tìm kiếm dữ liệu không ảnh hưởng đến trạng thái server.

38

Khi nào dùng POST

- Ưu điểm:
 - Bảo mật hơn phương thức GET
 - Không giới hạn dữ liệu truyền đi
- Nhược điểm:
 - Có thể gây ra lỗi nếu người dùng muốn quay lại trang kết quả (khi nhấn nút back, hoặc refresh) do bị expired
 - Dữ liệu truyền đi chậm hơn phương thức GET
- Dùng khi gửi dữ liệu nhạy cảm hoặc thay đổi trạng thái server (ví dụ: thêm mới, cập nhật dữ liệu, đăng nhập).

39

Giao thức phi trạng thái -Stateless Protocol

- Hypertext Transfer Protocol (HTTP) là **giao thức phi trạng thái (stateless)**: HTTP không lưu giữ trạng thái giữa các yêu cầu (request). Mỗi yêu cầu HTTP được xử lý như một yêu cầu độc lập, server không "nhớ" thông tin về các yêu cầu trước đó. Giao thức không trạng thái không yêu cầu máy chủ lưu giữ thông tin hoặc trạng thái về mỗi người dùng trong suốt thời gian của nhiều yêu cầu.
- Ví dụ: Khi truy cập trang "login.php", server không tự động biết rằng đã đăng nhập trước đó.
 - No shopping carts
 - No log-in

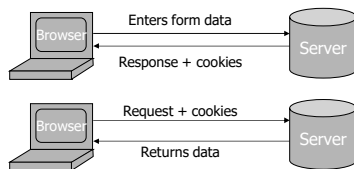
40

Giao thức phi trạng thái -Stateless Protocol

- Nhưng một số ứng dụng web có thể phải theo dõi tiến trình của người dùng từ trang này sang trang khác, ví dụ khi máy chủ web được yêu cầu để tùy chỉnh nội dung của trang web cho người dùng. Giải pháp cho những trường hợp này bao gồm:
 - việc sử dụng cookie HTTP.
 - phiên phía máy chủ (session),
 - các biến ẩn (khi trang hiện tại chứa một biểu mẫu) và
 - Viết lại URL bằng các tham số được mã hóa URI, ví dụ: `/index.php?session_id=some_unique_session_code`.

41

Giao thức phi trạng thái -Stateless Protocol



HTTP là stateless protocol (phi trạng thái); cookies bổ sung trạng thái

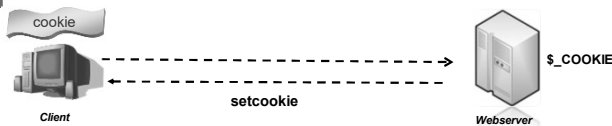
42

Truyền nhận dữ liệu

- Cookie: Lưu thông tin trạng thái ở phía client (trình duyệt).
- Session: Lưu thông tin trạng thái ở phía server, chỉ trao đổi session ID với client.
- Token-based authentication (JWT): Một cách hiện đại để lưu trạng thái người dùng trên client và server mà không cần sử dụng session.

43

Cookie - Giới thiệu



- Là 1 tập tin (thường là file text) được server **lưu xuống máy client**
- Mỗi lần client gửi request một trang web, đồng thời sẽ gửi kèm file cookie đã lưu lần trước lên server
- Việc xử lý **thông tin** (lưu, lấy) trong cookie **do server thực hiện**
- Thường được sử dụng để **lưu thông tin cá nhân của client**

44

Cookie - Giới thiệu

- Cookie được tạo ra bởi **website (ở server)** và gửi tới **browser**, do vậy 2 website khác nhau (dù cùng host trên 1 server) sẽ có **2 cookie** khác nhau gửi tới **browser**.
- Mỗi **browser quản lý và lưu trữ** cookie **theo cách riêng** của mình, cho nên **2 browser** cùng truy cập vào **1 website** sẽ nhận được **2 cookie** khác nhau.

45

Cookie - Giới thiệu

- Ví dụ về cookies
 - Name session-token
 - Content "s7yZiOvFm4YymG...."
 - Domain .amazon.com
 - Path /
 - Send For Any type of connection
 - Expires Monday, September 08, 2031 7:19:41 PM

46

Cookie – Cú pháp

- Lệnh ghi cookie
 - setcookie**(name, value, expire, path, domain);
 - **name**: Tên cookie
 - **value**: Giá trị cookie
 - **expire**: Thời điểm mà cookie hết hiệu lực
 - **path**: Đường dẫn trên server mà cookie có hiệu lực. Một ký tự dấu gạch chéo (/) cho phép Cookie có hiệu lực đối tất cả các thư mục.
 - **domain**: Xác định tên miền mà cookie có hiệu lực
 - Bắt buộc phải xuất hiện trước thẻ <html>
 - Có thể gọi nhiều lần để tạo nhiều cookie
 - VD: setcookie("Ten", "Sang", time()+100);

```
<?php
//Setting a cookie
setcookie("username","John Carter",time()+30*24*60*60);
?>
```

47

Cookie – Cú pháp

- Lấy giá trị cookie


```
echo $_COOKIE["cookieName"];
```
- Xóa cookie


```
setcookie("cookieName", "", time() -3600);
```
- Kiểm tra cookie đã được đặt hay chưa trước khi truy cập giá trị của nó, sử dụng hàm **isset()**

```
// kiểm tra cookies được cài đặt chưa
if(isset($_COOKIE["username"])){
    echo "Hi " . $_COOKIE["username"];
} else{
    echo "Welcome Guest!";
}
```

48

Ví dụ: trang a.php

```
<?php
    $t="1111";
    setcookie("a",$t);
?>
<html>
<head>
    <title>Trang a</title>
</head>
<body>
    gia tri ghi lên cookies:
    <?php
        echo $t;
    ?>
    <a href="b.php"> qua trang b </a>
</body>
</html>
```

49

Ví dụ: trang b.php

```
<html>
<head>
    <title>Trang b</title>
</head>
<body>
    <a href="a.php"> qua trang a </a>
    <?php
        if (isset($_COOKIE['a'])) {
            echo "gia tri lay duoc " . $_COOKIE['a'];
        }
        else
            echo "khong lay duoc";
    ?>
</body>
</html>
```

50

Ví dụ: kết quả

- Trang "a.php"



- Trang "b.php"



51

Nhược điểm của cookies

- Lưu trữ dữ liệu bằng cookie có vấn đề về **bảo mật**. Vì cookie được lưu trữ trên máy tính của người dùng nên kẻ tấn công có thể dễ dàng chen dữ liệu gây hại vào ứng dụng và có thể phá vỡ ứng dụng của người dùng.
- Khi trình duyệt yêu cầu một URL đến máy chủ, tất cả dữ liệu cookie cho một trang web sẽ tự động được gửi đến server trong yêu cầu, điều này ảnh hưởng đến **hiệu suất trang web**.

52

Session - Giới thiệu

- Là đoạn dữ liệu được **lưu trên server**, khi **browser** có **yêu cầu** lấy dữ liệu từ **session** thì **server** **cung cấp**.
- Mỗi **session** lưu **định danh duy nhất** cho **từng client**
- Mục đích lưu biến dữ liệu dùng chung cho nhiều trang trong 1 phiên làm việc của client

53

53

Session – Cú pháp

- Khởi động Session:
 - `session_start();`
 - **Bắt buộc phải xuất hiện trước thẻ <html>**
- Ghi giá trị Session
`$_SESSION["sessionVar"] = $value;`
- Đọc giá trị từ Session
 - Cách 1: `$value = $_SESSION["sessionVar"];`
 - Cách 2: `if (isset($_SESSION["sessionVar"]))
echo $_SESSION["sessionVar"];`
- Hủy biến trong Session
`unset($_SESSION["sessionVar"]);`
- Hủy cả Session
`session_destroy();`

54

54

Ví dụ: trang SessionA.php

```
<?php
    session_start();
    $_SESSION['username'] = 'guest';
    $_SESSION['password'] = '12345';
?>
<html>
<head>
    <title>Trang Session A</title>
</head>
<body>
    <a href="SessionB.php">Click để kiểm tra. </a>
</body>
</html>
```

55

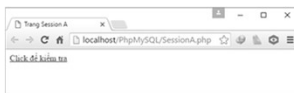
Ví dụ: trang SessionB.php

```
<?php
    session_start();
?>
<html>
<head><title>Trang Session B</title>
</head>
<body>
    Giá trị session lấy được
    username = <?php echo $_SESSION['username']; ?><br>
    time = <?php echo $_SESSION['password']; ?>
    <a href="SessionA.php">Quay lại trang A</a>
</body>
</html>
```

56

Kết quả

- Trang "SessionA.php"



- Trang "SessionB.php"



57

So sánh giữa Cookie và Session

- Cookie và Session đều có chung mục đích là lưu giữ data để truyền từ 1 trang web sang 1 trang web khác (trên cùng website).
- Phương thức lưu trữ và quản lý data của Cookie và Session có phần khác nhau.
- Cookie sẽ được lưu trữ tại browser, do browser quản lý và browser sẽ tự động truyền cookie ngược lên server mỗi khi truy cập vào 1 trang web trên server.
- Dữ liệu lưu trữ trong Session được webservice quản lý
- Browser chỉ truyền ID của session lên server để lấy dữ liệu khi cần

Đại học Khoa học và Công nghệ Hà Nội

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

UT-HCM

Phát triển ứng dụng Web

58

58

Sử dụng Cookie và Session

- Sử dụng Session hoặc Cookie là tùy vào lựa chọn của Lập trình viên, tuy nhiên Session thường được ưa chuộng hơn Cookie vì một số lý do sau:
 - Session vẫn sử dụng được trong trường hợp vùng nhớ Cookie bị chặn.
 - Lượng data truyền tải giữa browser và server: chỉ mỗi session ID được truyền giữa browser và server, data thực sự được website lưu trữ trên server.
 - Bảo mật: Càng ít thông tin được truyền tải qua lại giữa browser và client càng tốt, và càng ít thông tin được lưu trữ tại client càng tốt.

Đại học Khoa học và Công nghệ Hà Nội

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

UT-HCM

Phát triển ứng dụng Web

59

59

So sánh: Cookie – Session - Database

Đại học Khoa học và Công nghệ Hà Nội

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

UT-HCM

Phát triển ứng dụng Web

60

60

JWT

- **Header:** Chứa thông tin về thuật toán mã hóa (ví dụ: HMAC, SHA256).
- **Payload:** Chứa dữ liệu (claims) cần truyền tải, ví dụ: user_id, roles.
- **Signature:** Được tạo ra bằng cách mã hóa (HMAC SHA256) header và payload với một secret key.
- So với SESSION: JWT Không lưu trạng thái trên server (stateless). Giảm tải cho server, chỉ cần xác thực token. Có thể sử dụng mã hóa và ký số để bảo mật.
- Ứng dụng của JWT: Single Sign-On (SSO) cho phép user đăng nhập vào nhiều ứng dụng chỉ với một tài khoản duy nhất. User Authentication: JWT cung cấp khả năng xác thực người dùng và cấp quyền cho họ truy cập vào các tài nguyên mong muốn trong hệ thống...

64

JWT

- **Bước 1:** Server tạo JWT: Khi người dùng đăng nhập thành công, server tạo một JWT, mã hóa thông tin cần thiết (ví dụ: user_id, roles) và gửi token về client.
- **Bước 2:** Client lưu trữ JWT: JWT có thể được lưu trữ trên client bằng: HTTP-only cookie: Bảo mật tốt hơn vì không thể truy cập từ JavaScript hoặc localStorage hoặc sessionStorage: Phù hợp với ứng dụng SPAs (Single Page Applications).
- **Bước 3:** Client gửi JWT trong mỗi yêu cầu. Khi client gửi yêu cầu đến server, JWT được đính kèm trong HTTP Header và COOKIE
- **Bước 4:** Server xác thực JWT. Khi nhận được yêu cầu từ client, server: Lấy JWT từ header hoặc COOKIE. Xác thực chữ ký của token bằng secret key. Giải mã payload để truy xuất thông tin.

65

Q & A



Cảm ơn đã theo dõi

Hy vọng cùng nhau đi đến thành công.

66
