QUẢN LÝ TRẠNG THÁI NGƯỜI DÙNG VỚI COOKIES VÀ SESSIONS

Nội dung khóa học

Phần 1: Cookies trong PHP

Tìm hiểu chi tiết về cách tạo, quản lý và xóa cookies. Học cách thiết lập thời gian hết hạn, đường dẫn và bảo mật cho cookies.

Phần 2: Sessions trong PHP

Khám phá cách khởi tạo, lưu trữ và hủy sessions an toàn. Tìm hiểu về session_id, session handlers và cách xử lý dữ liệu người dùng giữa các trang.

Phần 3: So sánh và tích hợp

Phân tích sâu về thời điểm sử dụng cookies hoặc sessions. Học cách kết hợp cả hai để tạo trải nghiệm người dùng tối ưu.

Phần 1: Cookies trong PHP

Cookies là các tập tin văn bản nhỏ (thông thường dưới 4KB) được lưu trữ trên thiết bị của người dùng thông qua trình duyệt web. Chúng được tạo ra bởi máy chủ web và gửi đến trình duyệt của người dùng khi họ truy cập một trang web, nhằm lưu trữ thông tin trạng thái giữa các lần truy cập.

Mỗi cookie bao gồm các thành phần quan trọng như tên (name), giá trị (value), thời gian hết hạn (expiry date), đường dẫn (path) và domain. Khi người dùng quay lại trang web, trình duyệt sẽ tự động gửi cookies liên quan đến trang đó trong HTTP request header, cho phép máy chủ nhận diện người dùng và khôi phục trạng thái trước đó.

Cách hoạt động của Cookies

1

Người dùng truy cập website

Người dùng mở trình duyệt và truy cập vào một trang web được phát triển bằng PHP. Trình duyệt gửi HTTP request đến máy chủ.

2

Máy chủ tạo cookie

Mã PHP trên máy chủ xử lý request, tạo cookie với thông tin cần lưu trữ và gửi lại trong HTTP response header kèm theo nội dung trang web.

3

Lưu trữ cookie

Trình duyệt nhận response, hiển thị nội dung trang web và lưu cookie vào bộ nhớ tạm hoặc ổ đĩa của người dùng theo thời hạn được chỉ định.

4

Gửi lại cookie

Khi người dùng truy cập lại trang web, trình duyệt tự động gửi tất cả các cookie liên quan trong HTTP request header, cho phép máy chủ nhận diện người dùng và tùy chỉnh trải nghiệm.

Tạo và đọc Cookie trong PHP

Tạo cookie với hàm setcookie()

```
// Cú pháp cơ bản
setcookie(
    string $name,       // Tên cookie
    string $value = "", // Giá trị cookie
    int $expires_or_options = 0, // Thời gian hết hạn
    string $path = "",  // Đường dẫn cookie có hiệu lực
    string $domain = "", // Tên miền cookie có hiệu lực
    bool $secure = false, // Chỉ gửi qua HTTPS nếu true
    bool $httponly = false // Không cho phép JS truy cập nếu true
);

// Ví dụ đơn giản: Cookie hết hạn sau 1 giờ
setcookie("user", "NguyenVanA", time() + 3600);
```

Đọc giá trị cookie

```
// Kiểm tra sự tồn tại của cookie
if(isset($_COOKIE["user"])) {
    $user = $_COOKIE["user"];
    echo "Xin chào " . $user;
} else {
    echo "Cookie không tồn tại hoặc đã hết hạn";
}
```

Cookie được lưu trong biến toàn cục $_COOKIE dưới dạng một mảng liên kết, với khóa là tên cookie. Cookie chỉ có thể đọc được sau khi được gửi từ trình duyệt đến máy chủ trong yêu cầu HTTP tiếp theo.

Lưu ý: Giá trị cookie tự động được mã hóa URL khi gửi và giải mã khi nhận, nhưng không được mã hóa bảo mật.

Ví dụ ghi nhớ tùy chọn người dùng

Cách sử dụng cookie để nhớ màu yêu thích của người dùng.

```
// Lưu màu yêu thích vào cookie
if(isset($_POST['color'])) {
    $selected_color = $_POST['color'];
    setcookie("preferred_color", $selected_color, time() + (86400 * 30), "/");
    // Cookie có thời hạn 30 ngày
    echo "Đã lưu màu yêu thích của bạn: " . $selected_color;
}

// Hiển thị màu đã chọn từ cookie
$color = isset($_COOKIE['preferred_color']) ? $_COOKIE['preferred_color'] : "";
```

Form cho phép người dùng chọn màu yêu thích:

```
<form method="post" action="">
    <p>Chọn màu yêu thích:
        <input type="radio" name="color" value="red"> Đỏ
        <input type="radio" name="color" value="blue"> Xanh dương
        <input type="radio" name="color" value="green"> Xanh lá
    </p>
    <input type="submit" value="Lưu tùy chọn">
</form>
```

Xóa Cookie trong PHP

Để xóa cookie trong PHP, thiết lập lại cookie với thời gian hết hạn trong quá khứ, khiến trình duyệt tự động loại bỏ nó.

```
// Xóa cookie "user"
setcookie("user", "", time() - 3600);

// Xóa cookie với đường dẫn cụ thể
setcookie("user", "", time() - 3600, "/myapp/", "example.com", true);
```

Lưu ý: Các tham số path, domain, secure phải khớp với giá trị khi tạo cookie. Nếu không, trình duyệt sẽ coi đây là cookie khác và việc xóa sẽ thất bại. Giá trị time() - 3600 đặt thời gian hết hạn 1 giờ trước.

Ví dụ hoàn chỉnh: Xóa cookie

```
// Kiểm tra và xóa cookie
if(isset($_COOKIE["user"])) {
    $user = $_COOKIE["user"];
    setcookie("user", "", time() - 3600, "/");
    echo "Đã xóa cookie cho người dùng: " . $user;
} else {
    echo "Cookie không tồn tại hoặc đã bị xóa trước đó";
}
```

Đoạn mã trên kiểm tra sự tồn tại của cookie "user", xóa nó bằng cách đặt thời gian hết hạn vào quá khứ, và hiển thị thông báo xác nhận.

Ứng dụng thực tế của Cookies

Xác thực và quản lý phiên đăng nhập

Lưu trữ token xác thực bảo mật để người dùng không cần đăng nhập lại mỗi lần truy cập website. .

Duy trì trạng thái giỏ hàng thương mại điện tử

Lưu trữ và theo dõi danh sách sản phẩm trong giỏ hàng người dùng ngay cả khi họ chưa đăng nhập.

Cá nhân hóa trải nghiệm người dùng

Lưu trữ các tùy chọn giao diện và thiết lập cá nhân như chế độ sáng/tối, ngôn ngữ ưa thích, kích thước font chữ, bố cục trang, và các cài đặt hiển thị khác.

Phân tích hành vi và tối ưu hóa trang web

Thu thập dữ liệu về hành vi người dùng, thống kê lượt truy cập, thời gian lưu lại trang, và các chỉ số tương tác khác.

Phần 2: Sessions trong PHP

Khác với cookies, dữ liệu session được lưu trữ hoàn toàn trên máy chủ, giúp bảo vệ thông tin nhạy cảm như dữ liệu đăng nhập, thông tin cá nhân và giao dịch.

Sessions đặc biệt hữu ích cho các ứng dụng web cần duy trì trạng thái người dùng giữa các lượt truy cập với mức độ bảo mật cao.

Cách hoạt động của Sessions

1

Khởi tạo session

Khi hàm session_start() được gọi, PHP kiểm tra cookie PHPSESSID. Nếu chưa tồn tại, PHP sẽ tạo một session ID duy nhất và thiết lập cookie này trong trình duyệt người dùng.

2

Lưu trữ dữ liệu

Dữ liệu được lưu vào biến toàn cục $_SESSION trên máy chủ, thường được lưu trong các tệp tạm thời tại thư mục session.save_path hoặc trong cơ sở dữ liệu tùy theo cấu hình.

3

Truy xuất dữ liệu

Trong các request tiếp theo, PHP đọc session ID từ cookie PHPSESSID, sử dụng nó để định danh và khôi phục chính xác dữ liệu session của người dùng đó từ máy chủ.

4

Kết thúc session

Session có thể tự động hết hạn sau một khoảng thời gian (thiết lập qua session.gc_maxlifetime) hoặc được hủy chủ động thông qua hàm session_destroy() khi người dùng đăng xuất.

Sử dụng Sessions trong PHP

Khởi tạo và lưu trữ dữ liệu session

```
<?php
// Bắt đầu hoặc tiếp tục một session
session_start();

// Lưu trữ thông tin người dùng vào session
$_SESSION['username'] = 'nguyenvan';
$_SESSION['is_logged_in'] = true;

// Lưu trữ giỏ hàng như một mảng đa chiều
$_SESSION['cart'] = [
    'item1' => ['name' => 'Laptop', 'price' => 15000000],
    'item2' => ['name' => 'Chuột', 'price' => 500000]
];

// Lưu ý: Dữ liệu sẽ được giữ trên máy chủ cho đến khi session kết thúc
?>
```

Truy xuất và xử lý dữ liệu session

```
<?php
// Khởi tạo session - phải gọi trước khi truy cập $_SESSION
session_start();

// Kiểm tra sự tồn tại của biến session trước khi sử dụng
if(isset($_SESSION['username'])) {
    echo "Xin chào: " . $_SESSION['username'];
}

// Duyệt qua mảng đa chiều trong session
if(isset($_SESSION['cart'])) {
    echo "<h3>Giỏ hàng của bạn:</h3>";
    foreach($_SESSION['cart'] as $item) {
        echo $item['name'] . ": " .
            number_format($item['price']) . " VND<br>";
    }
}

// Gợi ý: Sử dụng session_destroy() để xóa toàn bộ dữ liệu session
?>
```

Lưu ý rằng session_start() phải được gọi trước khi bất kỳ dữ liệu nào được gửi đến trình duyệt, và mỗi trang sử dụng session đều phải gọi hàm này. Dữ liệu session được lưu trữ trên máy chủ, giúp bảo mật thông tin người dùng.

Ví dụ hệ thống đăng nhập với Session

1. Trang đăng nhập (login.php)

```
<?php
session_start();

// Kiểm tra nếu người dùng đã đăng nhập rồi
if(isset($_SESSION['logged_in']) && $_SESSION['logged_in'] === true) {
    header("Location: dashboard.php");
    exit;
}

// Kiểm tra form đăng nhập
if($_SERVER["REQUEST_METHOD"] == "POST") {
    // Thông tin đăng nhập mẫu (trong thực tế nên lấy từ database)
    $valid_username = "admin";
    $valid_password = "password123";

    // Kiểm tra thông tin đăng nhập
    if($_POST['username'] === $valid_username && $_POST['password'] === $valid_password) {
        // Đăng nhập thành công, tạo session
        $_SESSION['logged_in'] = true;
        $_SESSION['username'] = $_POST['username'];
        $_SESSION['login_time'] = time();

        // Chuyển hướng đến trang dashboard
        header("Location: dashboard.php");
        exit;
    } else {
        $error_message = "Tên đăng nhập hoặc mật khẩu không chính xác!";
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Đăng nhập</title>
</head>
<body>
    <h2>Đăng nhập hệ thống</h2>

    <?php if(isset($error_message)): ?>
        <p style="color: red;"><?php echo $error_message; ?></p>
    <?php endif; ?>

    <form method="post" action="">
        <div>
            <label>Tên đăng nhập:</label>
            <input type="text" name="username" required>
        </div>
        <div>
            <label>Mật khẩu:</label>
            <input type="password" name="password" required>
        </div>
        <div>
            <input type="submit" value="Đăng nhập">
        </div>
    </form>
</body>
</html>
```

2. Trang Dashboard (dashboard.php)

```
<?php
session_start();

// Kiểm tra xem người dùng đã đăng nhập chưa
if(!isset($_SESSION['logged_in']) || $_SESSION['logged_in'] !== true) {
    // Nếu chưa đăng nhập, chuyển hướng về trang đăng nhập
    header("Location: login.php");
    exit;
}

// Tính thời gian đã đăng nhập
$login_duration = time() - $_SESSION['login_time'];
$minutes = floor($login_duration / 60);
$seconds = $login_duration % 60;
?>

<!DOCTYPE html>
<html>
<head>
    <title>Dashboard</title>
</head>
<body>
    <h2>Chào mừng, <?php echo $_SESSION['username']; ?>!</h2>
    <p>Bạn đã đăng nhập trong: <?php echo $minutes; ?> phút <?php echo $seconds; ?> giây</p>

    <p>Đây là nội dung chỉ dành cho người dùng đã đăng nhập.</p>

    <a href="logout.php">Đăng xuất</a>
</body>
</html>
```

Trong ví dụ này, chúng ta sử dụng session để:

- Lưu trữ trạng thái đăng nhập của người dùng (`$_SESSION['logged_in']`)
- Lưu thông tin người dùng (`$_SESSION['username']`)
- Theo dõi thời gian đăng nhập (`$_SESSION['login_time']`)
- Kiểm tra quyền truy cập trang và chuyển hướng người dùng phù hợp

Hệ thống này có thể được mở rộng thêm với các tính năng như: xác thực hai lớp, ghi nhớ đăng nhập (remember me), giới hạn thời gian phiên làm việc, và nhiều tính năng bảo mật khác.

Hủy Session và Đăng xuất

Hủy một biến session

```
// Hủy một biến session cụ thể
unset($_SESSION['username']);

// Xác nhận biến đã bị hủy
if (!isset($_SESSION['username'])) {
    echo "Biến session đã bị hủy";
}
```

Khi chỉ cần xóa một thông tin cụ thể của người dùng, sử dụng hàm **unset()** để hủy biến session đó mà không ảnh hưởng đến toàn bộ session.

Hủy toàn bộ session (đăng xuất)

```
// Xóa tất cả dữ liệu session
$_SESSION = array();

// Hủy cookie session nếu có
if (isset($_COOKIE[session_name()])) {
    setcookie(session_name(), '', time()-42000, '/');
}

// Hủy session
session_destroy();

// Chuyển hướng về trang đăng nhập
header("Location: login.php");
exit();
```

Đăng xuất đúng cách cần xóa dữ liệu session, hủy cookie session và gọi **session_destroy()** để xóa hoàn toàn session khỏi máy chủ.

Phần 3: So sánh Cookies và Sessions

Tiêu chí	Cookies	Sessions
Vị trí lưu trữ	Máy khách (trình duyệt người dùng)	Máy chủ (server-side)
Dung lượng tối đa	Khoảng 4KB mỗi domain	Không giới hạn cứng (phụ thuộc tài nguyên server)
Thời gian tồn tại	Có thể thiết lập dài hạn (ngày/tháng/năm)	Thường ngắn hạn (hết hạn khi đóng trình duyệt)
Độ bảo mật	Thấp hơn (dễ bị đánh cắp, chỉnh sửa từ phía client)	Cao hơn (dữ liệu được bảo vệ trên server)
Tài nguyên máy chủ	Tiêu thụ ít (chủ yếu xử lý ở client)	Tiêu thụ nhiều (lưu trữ và xử lý trên server)
Phù hợp cho	Dữ liệu không nhạy cảm, tùy chọn người dùng, tracking	Thông tin đăng nhập, dữ liệu nhạy cảm, giỏ hàng

Ứng dụng kết hợp Cookies và Sessions

Đăng nhập và xác thực

Sessions lưu trữ trạng thái đăng nhập hiện tại và thông tin phiên làm việc an toàn. Cookies lưu trữ token xác thực "ghi nhớ đăng nhập" giúp người dùng không cần đăng nhập lại trong các lần truy cập tiếp theo.

Quản lý tùy chọn người dùng

Sessions quản lý các tùy chọn tạm thời trong phiên làm việc hiện tại. Cookies lưu trữ các tùy chọn dài hạn như ngôn ngữ ưa thích, chế độ hiển thị tối/sáng, và bố cục giao diện người dùng.

Giỏ hàng thông minh

Sessions bảo vệ dữ liệu giỏ hàng cho người dùng đã đăng nhập, đảm bảo tính riêng tư. Cookies duy trì thông tin giỏ hàng cho người dùng chưa đăng nhập, tạo trải nghiệm mua sắm liền mạch ngay cả khi chuyển thiết bị.

Bảo mật đa lớp

Sessions quản lý quyền truy cập và phân quyền người dùng trong thời gian thực. Cookies hỗ trợ bảo mật nâng cao thông qua việc lưu trữ token xác thực hai yếu tố, dấu vân tay thiết bị và thông tin phát hiện bất thường.