Session

Session và HTTP Session Trong Lập Trình Java Servlets

1. Giới thiệu về Session:

- **Định nghĩa và vai trò**: Trong phát triển ứng dụng web, session đóng vai trò cực kỳ quan trọng. Nó là một cách để lưu trữ thông tin người dùng qua các yêu cầu HTTP khác nhau. Điều này cho phép tạo ra một trải nghiệm người dùng liền mạch và an toàn, như giữ người dùng đăng nhập khi họ di chuyển từ trang này sang trang khác.
- **Cách hoạt động**: Session hoạt động bằng cách tạo ra một ID session duy nhất, thường được lưu trữ trong cookie trên trình duyệt người dùng. Mỗi khi người dùng gửi một yêu cầu đến server, ID session này được gửi kèm theo, giúp server nhận diện và duy trì trạng thái cho người dùng đó.

2. HTTP Session trong Java Servlets:

- **Giới thiệu về HTTP session**: HTTP session trong Java Servlets là một cách triển khai của session được quản lý thông qua API Servlet. Nó cho phép lập trình viên lưu trữ dữ liệu trạng thái người dùng qua nhiều yêu cầu HTTP.
- **Phương pháp tạo, quản lý và hủy session**: Trong Java Servlets, một session được tạo thông qua `HttpServletRequest.getSession()` và có thể được quản lý bằng cách thêm, sửa, hoặc xóa các thuộc tính. Session có thể được hủy bằng `HttpSession.invalidate()`.

3. Lập trình với HTTP Session:

- **Hướng dẫn lập trình**: Để làm việc với HTTP session, bạn cần tạo một session (nếu chưa có) và sau đó làm việc với các thuộc tính trong session đó. Điều này thường được thực hiện trong các servlet hoặc JSP.
- **Ví dụ cụ thể**: Giả sử bạn muốn lưu tên người dùng trong session. Bạn có thể làm điều này bằng cách:

```
HttpSession session = request.getSession();
session.setAttribute("username", "NguyenVanA");
```

Phương thức

1. `getId()`**:**

- **Mô tả**: Phương thức này trả về ID duy nhất của session. Đây là một chuỗi được tạo ra tự động khi session được tạo.
- **Ứng dụng**: Sử dụng để theo dõi hoặc log session của người dùng. Ví dụ: "Session ID 12345 đã được tạo."

```
HttpSession session = request.getSession();
String sessionId = session.getId();
System.out.println("Session ID: " + sessionId);
```

2. `getCreationTime()`**:**

- **Mô tả**: Trả về thời gian (được tính bằng milliseconds từ epoch) mà session được tạo ra.
- **Ứng dụng**: Hữu ích trong việc theo dõi tuổi của session, giúp quyết định khi nào nên hủy session.

```
HttpSession session = request.getSession();
long creationTime = session.getCreationTime();
```

```
System.out.println("Session Creation Time: " + new Date(creat
```

3. `getLastAccessedTime()`:

- **Mô tả**: Trả về thời gian cuối cùng mà session này được truy cập.
- **Ứng dụng**: Cung cấp thông tin về mức độ hoạt động của người dùng, có thể hữu ích cho việc timeout session.

```
HttpSession session = request.getSession();
long lastAccessed = session.getLastAccessedTime();
System.out.println("Last Accessed Time: " + new Date(lastAcce
```

4. `setMaxInactiveInterval(int interval)`:

- **Mô tả**: Đặt thời gian tối đa (tính bằng giây) mà session có thể không hoạt động trước khi nó được hủy.
- **Ứng dụng**: Quan trọng cho việc quản lý tài nguyên và bảo mật, đặc biệt là trong việc ngăn chặn session hi-jacking.

```
HttpSession session = request.getSession();
// Set to 15 minutes
session.setMaxInactiveInterval(15 * 60);
```

5. `getMaxInactiveInterval()`:

- **Mô tả**: Trả về thời gian tối đa (tính bằng giây) mà session có thể không hoạt động trước khi bị hủy.
- **Ứng dụng**: Hữu ích khi cần kiểm tra hoặc log thông tin cấu hình session.

```
HttpSession session = request.getSession();
int interval = session.getMaxInactiveInterval();
System.out.println("Max Inactive Interval: " + interval + " s
```

6. `invalidate()`:

- **Mô tả**: Hủy session, loại bỏ bất kỳ đối tượng nào được lưu trữ trong nó.

- **Ứng dụng**: Rất quan trọng trong việc quản lý session, đặc biệt là khi người dùng đăng xuất.

```
HttpSession session = request.getSession();
// When user logs out
session.invalidate();
```

7. `isNew()`:

- **Mô tả**: Kiểm tra xem session này có phải là mới tạo hay không.
- **Ứng dụng**: Hữu ích trong việc xác định liệu người dùng có đang tiếp tục từ một session cũ hay bắt đầu một session mới.

```
HttpSession session = request.getSession();
if (session.isNew()) {
    System.out.println("This is a new session.");
} else {
    System.out.println("Welcome back!");
}
```