

RECENT POSTS

Hội chứng ITB và cách khắc
phục của cá nhân

Tổng kết năm 2018. chào đón

Session và HttpSessionListener

Ten ten ten tên!

Hi hi hi. Chào mọi người! Rất hân hạnh được gặp các bạn tại blog của mình.

Bài viết này mình sẽ trình bày về HttpSessionListener và ứng dụng của nó vào thực tế.

Đầu tiên mình sẽ nói lại về định nghĩa session.

Session nói chung và trong **jsp** hay **servlet** nói riêng là 1 **interface** được cung cấp để lưu trữ các thông tin trong 1 phiên làm việc của bạn trên trang web. Phiên làm việc ở đây có thể hiểu là một quá trình thao tác của bạn từ lúc truy cập vào 1 trang web để mua hàng đến khi bạn thoát khỏi trang đó.

Trong **jsp** thì đối tượng **session** được khai báo 1 cách **implicit** tức là nó nằm trong nhóm những thành phần có sẵn của 1 trang jsp bạn chỉ việc sử dụng nó bằng chính tên session. Còn trong servlet thì bạn phải get nó từ **Request** và cú pháp như sau

`HttpSession session = request.getSession();`

Tiếp theo là 1 số thuộc tính và phương thức quan trọng của session. Ở đây mình chỉ nói những thuộc tính và phương thức hay dùng tới.

`session.invalidate()` — hủy session và hủy các đối tượng liên quan tới session

`session.setAttribute(name,value)` — hàm này dùng để set 1 biến và giá trị của nó vào session ví

dụ: `session.getAttribute("user","loga");`

`session.getAttribute(name)` — hàm lấy về giá trị của 1 biến ví dụ `session.getAttribute("user");`

`session.setMaxInactiveInterval(khoảng thời gian)` — hàm này

set 1 khoảng thời gian kể từ request cuối cùng từ client tới server cho đến khi session invalidate, khoảng thời gian ở đây tính bằng giây. Ví dụ bạn `setMaxInactiveInterval(10)` thì tính từ lần request cuối cùng từ client tới server nếu không có 1 request nào mới trong khoảng thời gian 10s này thì session sẽ invalidate, nhưng nếu trong khoảng thời gian 10 s đó có 1 request từ client tới server thì session sẽ tự động gọi `setMaxInactiveInterval(10)` ;

Trên đây đó là những gì cơ bản nhất về session. Tiếp theo mình sẽ đi vào `HttpSessionListener`.

Tại sao tui phải quan tâm tới `HttpSessionListener`? Câu trả lời là: Khi làm việc với session bạn muốn bắt sự kiện khi session được tạo ra và khi session được kết thúc. Ví dụ bạn muốn đếm số người đang online trên trang web của bạn thì bạn sẽ nghĩ là: "à, mình phải biết được khi nào session được tạo ra thì mình tăng biến đếm lên 1, còn khi session bị hủy(invalidate) thì giảm biến đếm 1, tất nhiên biến đếm này phải là biến static trong 1 class nào đó". Để làm được như vậy thì mình dùng tới `HttpSessionListener` (ai có cách khác xin chỉ giáo). Vậy thì `HttpSessionListener` có gì mà làm được thế?

Thưa, `HttpSessionListener` là 1 interface có 2 method sau:

- 1.`sessionCreated()` — bắt sự kiện khi session được tạo ra
- 2.`sessionDestroyed()` — bắt sự kiện khi session kết thúc(`invalidate`)

Ừ mà cái `HttpSessionListener` là 1 interface thì làm sao mà gọi được 2 cái hàm trên được nhỉ. Rất đơn giản bạn chỉ cần tạo 1 class mới implement `HttpSessionListener` là bạn có thể kiểm soát được khi nào thì `sessionCreated`, khi nào thì `sessionDestroyed`.

Chú ý: Để sử dụng được `HttpSessionListener` bạn nhớ thêm

```
<listener>
    <listener-class>MySessionLister</listener-class>
</listener>
```

vào trong file **web.xml**, **MySessionListener** là 1 class implement interface **HttpSessionListener**. À, còn 1 điều lưu ý nữa là trong file xml bạn có thể chỉnh thời gian timeout cho 1 session (tương tự **setMaxInactiveInterval**) như sau:

```
<session-config>
<session-timeout>
30
</session-timeout>
</session-config>
```

Vậy thì ứng dụng HttpSession vào thực tế như thế nào, làm cái cho dễ hiểu thử coi? Okay dưới đây mình trình bày ví dụ đếm số người online(thực chất là đếm số session, tuy nhiên sẽ không chính xác lắm nếu bạn test với 1 máy mở nhiều hơn 1 trình duyệt khác nhau) trên 1 trang web cực kỳ đơn giản để các bạn có thể hiểu thêm về session và HttpSessionListener.

Đầu tiên bạn tạo 1 class MySessionListener như sau

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package listener;

import java.util.Date;
import javax.servlet.http.HttpSession;
```

```
import javax.servlet.http.HttpSessionEvent;
import javax.servlet.http.HttpSessionListener;

/**
 *
 * @author loga
 */
public class MySessionListener implements
HttpSessionListener{

    private static int songuoionline=0;

    //ham de lay bien so nguoi online
    public static int getSonguoionline() {
        return songuoionline;
    }

    public void sessionCreated(HttpSessionEvent se) {
        HttpSession session = se.getSession(); //lay session từ
        HttpSessionEvent
        Date now = new Date();
        System.out.println("Da tao session co ID "+ session.getId()+"
        vao thoi diem " + now );
        System.out.println("Dang có " + songuoionline +" online");
        songuoionline ++;
    }

    public void sessionDestroyed(HttpSessionEvent se) {
        HttpSession session = se.getSession(); //lay session từ
```

```

HttpSession session = se.getSession(); // lấy session từ
HttpSessionEvent
Date now = new Date();
System.out.println("Đã hủy session có ID "+ session.getId()+"
vào thời điểm " + now );
songuoionline--;
}

}

```

Sau đó trong file web.xml add thêm đoạn code này vào:

```

<listener>
<listener-class>listener.MySessionListener</listener-class>
</listener>

```

Ta có file **web.xml** hoàn chỉnh như sau:

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
xmlns="http://java.sun.com/xml/ns/javaee&#8221;
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance&#8221;
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"&gt;
<session-config>
<session-timeout>
30
</session-timeout>
</session-config>

```

```
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
<listener>
<listener-class>listener.MySessionListener</listener-class>
</listener>
</web-app>
```

Chú ý: listener là 1 tên package chứa class MySessionListener

Tiếp đến tạo trang **demsonguoionline.jsp** như sau:

```
<%-
Document : songuoionline
Created on : Jun 22, 2009, 10:06:32 PM
Author : loga
-%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="listener.MySessionListener" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Demo HttpSessionListener</title>
</head>
```

```
<body>
<h2>Hiện có <%=MySessionListener.getSonguoionline()%>
online</h2>
</body>
</html>
```

Thử chạy coi nào, bạn hãy thử 2 đến 3 trình duyệt khác nhau tương ứng 3 user trên máy. Cái này gọi là so nguoi online vẫn chưa đúng lắm. Để đếm số người online đúng hơn thì bạn nên kiểm tra ip để tránh tình trạng 1 ip có nhiều hơn 1 session khi đó biến đếm của chúng ta sẽ không chính xác. Tuy nhiên đây chỉ là 1 ví dụ minh họa đơn giản để các bạn có thể hiểu rõ hơn về session cũng như HttpSessionListener.

Link demo <http://mega.1280.com/file/ASK8MKPS/> [*this link is currently unavailable*]

Tham khảo:

<http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/HttpSession.html>

<http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/HttpSession.html>

<http://www.roseindia.net/servlets/ListenerSession.shtml>

Chúc các bạn thành công!

📅 February 14, 2014 👤 longkyanh 📁 java 💡 httpsessionlistener, java, session

Leave a Reply

Your email address will not be published. Required fields are marked *

COMMENT

NAME *

EMAIL *

WEBSITE

POST COMMENT

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

PREVIOUS

So sanh aggregation với Composition khi thiet ke class diagram

NEXT

So sánh 2 phương thức Post, Get

Proudly powered by [WordPress](#)