

००० सेशन्स इन जावा सर्वलेट्स

मुझे आपकी प्रश्न के बारे में बताने के लिए HttpSession का काम कैसे चलता है, कैसे एक सेशन का मान सेट किया जाता है, और कैसे इसे प्राप्त किया जाता है। मैं “`session.setAttribute()`” में आपकी सवाल में एक टाइपो मानता हूँ और यह है कि आपने HttpSession का मतलब है, क्योंकि यह javax.servlet पैकेज में सही क्लास नाम है।

_____ कैसे काम करता है?

`HttpSession` एक इंटरफेस है `javax.servlet.http` पैकेज में जो जावा वेब एप्लिकेशन में स्टेट मैनेजमेंट को संभव बनाता है। मूल रूप से स्टेटलेस है—एक क्लाइंट (जैसे, एक ब्राउज़र) से एक सर्वर तक की प्रत्येक अनुरोध स्वतंत्र है, पूर्ववर्ती इंटरैक्शंस की कोई याद नहीं है। इस सीमितता को पार करने और उपयोगकर्ता-विशिष्ट डेटा को कई अनुरोधों के माध्यम से ट्रैक करने के लिए (जैसे, एक उपयोगकर्ता की वेबसाइट पर यात्रा के दौरान), `HttpSession` एक मैकेनिज्म प्रदान करता है “सेशन” को बनाए रखने के लिए।

यह इस तरह काम करता है:

- सेशन निर्माण:** जब एक उपयोगकर्ता पहली बार एक वेब एप्लिकेशन में एक सर्वलेट तक पहुंचता है, तो सर्वलेट कंटेनर (जैसे, टोमकैट) एक नया HttpSession ऑब्जेक्ट बनाता है। इस सेशन को एक अनूठा पहचानकर्ता सेशन आईडी दिया जाता है।
 - सेशन आईडी ट्रांसमिशन:** सेशन आईडी को क्लाइंट के ब्राउज़र को भेजा जाता है, आमतौर पर एक कुकी के रूप में नामित JSESSIONID। अगले अनुरोधों में, ब्राउज़र इस सेशन आईडी को शामिल करता है, जिससे सर्वर अनुरोध को मौजूदा सेशन से जोड़ सकता है।
 - फॉलबैक मैकेनिज्म:** अगर ब्राउज़र में कुकी डिसएबल्ड हैं, तो सर्वलेट कंटेनर ००० रीवाइटिंग का उपयोग एक फॉलबैक के रूप में कर सकता है। इस मामले में, सेशन आईडी को ००० में जोड़ा जाता है (जैसे, <http://example.com/page;jsessionid=abc123>), हालांकि यह अनुप्रयोग कोड में स्पष्ट समर्थन की आवश्यकता होती है।
 - सर्वर-साइड स्टोरेज:** वास्तविक सेशन डेटा (अट्रिब्यूट्स) सर्वर पर, नहीं क्लाइंट पर, स्टोर किया जाता है। क्लाइंट केवल सेशन आईडी रखता है, जिससे सेशन को संवेदनशील डेटा को स्टोर करने के लिए कुकी से अधिक सुरक्षित बनाता है। डेटा आमतौर पर सर्वर मेमोरी में रखा जाता है, लेकिन उन्नत कॉन्फिगरेशन में डिस्क या डेटाबेस में पर्सिस्ट किया जा सकता है।
 - सेशन लाइफसाइक्ल:** सेशन में एक टाइमआउट अवधि होती है (जैसे, 30 मिनट डिफॉल्ट, web.xml या प्रोग्रामेटिक रूप से कॉन्फिगर करने के माध्यम से कॉन्फिगर किया जा सकता है)। अगर उपयोगकर्ता इस समय से अधिक निष्क्रिय रहता है, तो सेशन समाप्त हो जाता है और उसका डेटा फेंक दिया जाता है। आप एक सेशन को भी मैन्युअल रूप से समाप्त कर सकते हैं, जैसे कि लॉगआउट के दौरान।

इस मैकेनिज्म के माध्यम से, सर्वर उपयोगकर्ता-विशिष्ट जानकारी, जैसे लॉगिन स्थिति या शॉपिंग कार्ट सामग्री, कई अनुरोधों के माध्यम से "यात्र रख सकता है।

सेशन का मान कैसे सेट किया जाता है

एक HttpSession में डेटा स्टोर करने के लिए, आप setAttribute विधि का उपयोग करते हैं। यह विधि एक कुंजी (एक String) और एक मान (किसी भी जावा ऑब्जेक्ट) को एक साथ जोड़ती है। यह कैसे किया जाता है:

- सेशन का ऑब्जेक्ट प्राप्त करें:** एक सर्वलेट में, HttpServletRequest ऑब्जेक्ट से HttpSession प्राप्त करें `request.getSession()`। यह विधि एक नया सेशन बनाती है अगर कोई मौजूद नहीं है या मौजूदा सेशन लौटाती है।
- अट्रिब्यूट सेट करें:** `setAttribute(key, value)` को HttpSession ऑब्जेक्ट पर कॉल करें।

यह एक सर्वलेट में एक उदाहरण है:

```
import javax.servlet.http.*;
import java.io.*;

public class SetSessionServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {
        // ...
        HttpSession session = request.getSession();
        // ...
        session.setAttribute("username", "Alice");
        // ...
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(" : username = Alice");
    }
}
```

इस कोड में:- `request.getSession()` एक सेशन उपलब्ध हो जाए। - `session.setAttribute("username", "Alice")` स्ट्रिंग "Alice" को कुंजी "username" के तहत स्टोर करता है।

सेशन का मान कैसे प्राप्त किया जाता है

सेशन से एक मान प्राप्त करने के लिए, `getAttribute` विधि का उपयोग करें। क्योंकि यह एक Object लौटाता है, आपको इसे सही प्रकार में कास्ट करना होगा। यह प्रक्रिया है:

- सेशन का ऑब्जेक्ट प्राप्त करें:** `request.getSession()` या `request.getSession(false)` का उपयोग करें (पहले से मौजूद सेशन लौटाता है या कोई सेशन मौजूद नहीं है तो null लौटाता है, नया सेशन बनाने से बचता है)।
- अट्रिब्यूट प्राप्त करें:** `getAttribute(key)` को कॉल करें और परिणाम को कास्ट करें।

यह एक उदाहरण है:

```

import javax.servlet.http.*;
import java.io.*;

public class GetSessionServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {
        //           (           `null`      )
        HttpSession session = request.getSession(false);

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        if (session != null) {
            //
            String username = (String) session.getAttribute("username");

            if (username != null) {
                out.println("      : username = " + username);
            } else {
                out.println("      username      .");
            }
        } else {
            out.println("      .");
        }
    }
}

```

इस कोड में: - `request.getSession(false)` मौजूदा सेशन प्राप्त करता है या `null` लौटाता है। - `session.getAttribute("username")` मान को प्राप्त करता है, `String` में कास्ट किया जाता है। - `null` चेक्स कोड में त्रुटियों से बचने के लिए हैं अगर सेशन या अट्रिब्यूट मौजूद नहीं है।

प्रैक्टिकल उदाहरण

यह एक संयुक्त उदाहरण है जो सेशन मान को सेट और प्राप्त करने को दिखाता है अनुरोधों के माध्यम से:

1. `http://localhost:8080/SessionExample.setSession` (जैसे, लॉगिन के लिए):

```

import javax.servlet.http.*;
import java.io.*;

```

```

public class SetSessionServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {
        HttpSession session = request.getSession();
        session.setAttribute("user", "Bob");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(" : Bob");
    }
}

```

2. सेशन सेटिंग को प्राप्त करने का एक सुरक्षित पेज (जैसे, एक सुरक्षित पेज के लिए):

```

import javax.servlet.http.*;
import java.io.*;

public class GetSessionServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {
        HttpSession session = request.getSession(false);

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        if (session != null && session.getAttribute("user") != null) {
            String user = (String) session.getAttribute("user");
            out.println(" , " + user);
        } else {
            out.println(" ");
        }
    }
}

```

जब एक क्लाइंट SetSessionServlet तक पहुंचता है, तो सेशन "Bob" को स्टोर करता है। बाद में, GetSessionServlet तक पहुंचने पर, इसे प्राप्त और प्रदर्शित करता है, क्योंकि सेशन अनुरोधों के माध्यम से बने रहता है।

अतिरिक्त नोट्स

- **सेशन टाइमआउट:** session.setMaxInactiveInterval(seconds) (जैसे, session.setMaxInactiveInterval(1800) 30 मिनट के लिए) या web.xml में सेट किया जा सकता है:

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

- **सेशन समाप्त करना:** session.invalidate() (जैसे, लॉगआउट के लिए) का उपयोग करें।
 - **सुरक्षा:** सेशन आईडी को हस्तांतरण से बचाने के लिए **क्रिप्टो** का उपयोग करें। अत्यधिक संवेदनशील डेटा को बिना एन्क्रिप्शन के स्टोर करने से बचें।
 - **टाइप कास्टिंग:** हमेशा getAttribute परिणाम को कास्ट करें, क्योंकि यह Object लौटाता है।
-

सारांश

- **कैसे काम करता है:** HttpSession उपयोगकर्ता स्टेट को एक सेशन आईडी (कुकी या **रीवाइटिंग** के माध्यम से) का उपयोग करके बनाए रखता है और डेटा को सर्वर पर स्टोर करता है।
- **एक मान सेट करें:** session.setAttribute("key", value)।
- **एक मान प्राप्त करें:** String value = (String) session.getAttribute("key").

यह आपको javax.servlet में HttpSession के बारे में एक स्पष्ट समझ देगा! अगर आपको और कोई स्पष्टीकरण चाहिए, तो मुझे बताएं!