



openHPI: Web-Technologien Session-Management

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Session-Management

Viele Web-Anwendungen müssen sich an frühere Interaktionen zwischen Browser und WWW-Server erinnern „können“, z.B.

- virtuelle Warenkörbe beim Online-Shopping,
- Webseiten mit Login, usw.

Aber HTTP ist **zustandsloses Protokoll**. Deshalb braucht es Konzepte zur Bildung von „**Sessions**“:

- Zusammenfassung zusammengehörenden Nutzerinteraktionen (Request-Response-Zyklen) zwischen Client und Server
- Sessions werden vermittelt einer **Session-ID** identifiziert, die übertragen werden können durch
 - **Cookies** – Key-Value-Paare im HTTP-Header (→ Woche 1)
 - „**Hidden Fields**“ in HTML-Formularen
 - **URL Rewriting**

Session-Management: URL Rewriting

URL Rewriting ist sehr simple, aber unzureichende Methode der Zustandslosigkeit von HTTP zu begegnen

- Beispiel `http://example.org/index.html?sessionid=1234567890`

Vorteile

- Einfach zu realisieren
- Session-Informationen für Nutzer einfach in der URL ablesbar

Nachteile

- Länge von URLs ist begrenzt
- da bei unverschlüsselter Verbindung die URL einfach mitgelesen werden kann, ist URL Rewriting nicht geeignet für sensitive Daten, z.B.: `http://example.org/login/?username=openhpi&password=12345`
- Session kann bei Kenntnis der URL durch Dritte einfach übernommen (gekapert) werden

Session-Management: Hidden Fields

Hidden Fields sind versteckte Felder in HTML-Formularen

- Formular besitzt Felder vom Typ **hidden**, für einen Namen und einen Wert, z.B. Session-ID
- Abgeschickt werden diese Formulare mit HTTP POST-Methode

```
<form name="session" method="post"
      action="dashboard.html">
  <input type="hidden"
        name="SessionID"
        value="1234567890" />

  <input type="submit"
        value="Zum Dashboard">
</form>
```

Vorteil

- Bei Übertragung mit HTTPS bleiben Session-Informationen geheim

Nachteile

- Session-Informationen sind im Quelltext der Seite auslesbar, wenn Seite unverschlüsselt übertragen wird
- Einfache Links so nicht möglich, da gesamte Navigation über Formulare realisiert werden muss

Session-Management: Cookies (1/2)

Cookie-Mechanismus (Erinnerung Woche 1):

1. HTTP-Server beauftragt Client, Cookie anzulegen, indem er im HTTP response das Headerfeld **set-cookie** setzt
 2. Browser speichert Cookie in internen Datenbank
 3. Bei jedem neuen Request an gleiche Domain sendet Browser all seine dazu passenden Cookies im Request-Headerfeld **cookie**
-
- Typischerweise werden nur (Session) IDs in Cookies gesendet, eigentliche Informationen (z.B. virtueller Warenkorb) werden serverseitig abgelegt

Session-Management: Cookies (2/2)

Cookie-Mechanismus (Erinnerung Woche 1):

Vorteile

- Bei Übertragung mit HTTPS bleiben Informationen geheim
- Einfache Wartbarkeit für Webseitenbetreiber und –Entwickler
- Session-Handling unabhängig von HTML

Nachteil

- Session-Informationen aus HTTP-Header auslesbar, wenn diese unverschlüsselt übertragen wird
- Tracking von Nutzer-Interaktionen sehr einfach möglich für Webseitenbetreiber und Dritte (Google, Facebook, ...)