ILOVA HOLATI. COOKIE-FAYLLAR. SEANSLAR

Yusupov Mirsaid Abdulaziz o'g'li
Farg'ona Davlat Universiteti Amaliy matematika va informatika kafedrasi o'qituvchisi
mirsaidbeky@gmail.com
Soliyeva Xurshida Tavakaljon qizi
Farg'ona Davlat Unversiteti, talaba
xurshidasoliyeva27@gmail.com

Anotatsiya : *Ushbu maqola web dasturlash jarayonida ilova holatini boshqarishning nazariy va amaliy asoslarini o'rganishga bag'ishlangan. HTTP protokolining holatsiz (stateless) xususiyati sababli, server va mijoz o'rtasidagi uzluksiz aloqani ta'minlash uchun qo'shimcha mexanizmlar — cookie-fayllar va seanslar zarur bo'ladi. Maqolada cookie-fayllar va seanslarning ishlash mexanizmlari, ularning o'zaro farqlari, xavfsizlik masalalari va zamonaviy web ilovalarda qo'llanishi atroflicha yoritilgan. Bundan tashqari, PHP, ASP.NET, Java va boshqa texnologiyalarda seans boshqaruvi bo'yicha kod misollari va qo'llanmalar berilgan. Ushbu ish dasturchilar, talabalar va IT mutaxassislari uchun foydali qo'llanma bo'lib xizmat qiladi.*

Kalit so'zlar : *Ilova holati, cookie-fayllar, seanslar, HTTP protokoli, HTTP so'rovlar, HTTP javoblar, foydalanuvchi autentifikatsiyasi, sessiya boshqaruvi, xavfsizlik protokollari, token asosidagi autentifikatsiya, seans identifikatori, seans muddati, foydalanuvchi ma'lumotlarini saqlash, server holatini boshqarish, mijoz holatini boshqarish, web arxitektura, HTTP headerlar, JavaScript cookie boshqaruvi, ASP.NET sessiyalar, PHP sessiyalar, Java Servlet sessiyalar, OAuth 2.0, JWT (JSON Web Token), CSRF hujumlaridan himoya, XSS hujumlaridan himoya*

Abstract : *This article is devoted to studying the theoretical and practical foundations of state management in web development. Due to the stateless nature of the HTTP protocol, additional mechanisms — such as cookies and sessions — are required to maintain continuous interaction between the server and the client. The article provides an in-depth analysis of the operation of cookies and sessions, their differences, security issues, and their application in modern web applications. Additionally, code examples and tutorials are provided for session management in PHP, ASP.NET, Java, and other technologies. This work serves as a useful guide for developers, students, and IT specialists.*

Keywords: *Application state, cookies, sessions, HTTP protocol, HTTP requests, HTTP responses, user authentication, session management, security protocols, token-based authentication, session identifier, session timeout, user data storage, server-side state management, client-side state management, web architecture, HTTP headers, JavaScript cookie management, ASP.NET sessions, PHP sessions, Java Servlet sessions, OAuth 2.0, JWT (JSON Web Token), CSRF protection, XSS protection.*

Аннотация: *Данная статья посвящена изучению теоретических и практических основ управления состоянием в веб-разработке. Из-за безсостояния протокола HTTP для поддержания непрерывного взаимодействия между сервером и клиентом необходимы дополнительные механизмы — такие как cookie-файлы и сессии. В статье подробно анализируется работа cookie-файлов и сессий, их различия, вопросы безопасности и применение в современных веб-приложениях. Кроме того, приводятся примеры кода и инструкции по управлению сессиями в PHP, ASP.NET, Java и других технологиях. Данная работа служит полезным руководством для разработчиков, студентов и IT-специалистов.*

Ключевые слова: *Состояние приложения, cookie-файлы, сессии, протокол HTTP, HTTP-запросы, HTTP-ответы, аутентификация пользователя, управление сессиями, протоколы безопасности, аутентификация на основе токенов, идентификатор сессии, время жизни сессии, хранение данных пользователя, управление состоянием сервера, управление состоянием клиента, веб-архитектура, заголовки HTTP, управление cookie-файлами с помощью JavaScript, сессии ASP.NET, сессии PHP, сессии Java Servlet, OAuth 2.0, JWT (JSON Web Token), защита от CSRF-атак, защита от XSS-атак.*

Kirish

Web dasturlash sohasida ilova holatini boshqarish bugungi kunda har bir dinamik va interaktiv web ilovasining asosiy talablaridan biridir. HTTP protokoli, o‘zining holatsiz (stateless) xususiyatiga ega bo‘lib, mijoz va server o‘rtasidagi har bir so‘rov va javobni alohida, mustaqil holatda ko‘rib chiqadi. Bu esa uzluksiz aloqani ta'minlash uchun qo‘shimcha mexanizmlar, ya'ni cookie-fayllar va seanslarning joriy etilishini zarur qiladi. Ilova holatini boshqarish, foydalanuvchi tajribasini (UX) yaxshilash, xavfsizlikni ta'minlash va samarali tizimni yaratish uchun muhimdir.Cookie-fayllar va seanslar web ilovalari uchun juda keng qo‘llaniladigan mexanizmlar bo‘lib, ular foydalanuvchi ma'lumotlarini saqlash, autentifikatsiya qilish va ma'lum bir vaqt davomida foydalanuvchi holatini boshqarish imkoniyatini beradi. Cookie-fayllar, foydalanuvchining brauzerida saqlanadigan kichik ma'lumotlarni o‘z ichiga oladi, seanslar esa serverda saqlanadigan ma'lumotlar to‘plamidan tashkil topadi. Ularning har biri o‘zining afzalliklari va kamchiliklariga ega bo‘lib, ularni to‘g‘ri va samarali ishlatish web dasturlashning eng muhim jihatlaridan biridir.Bundan tashqari, bu mexanizmlar xavfsizlik masalalari bilan chambarchas bog‘liqdir. Xavfsizlik protokollari, token asosidagi autentifikatsiya, seans identifikatorlari va sessiya muddati kabi

tushunchalar cookie-fayllar va seanslar bilan bog‘liq bo‘lib, ularni to‘g‘ri boshqarish orqali xavfsizlik darajasini oshirish mumkin. Xulosa qilib aytganda, ilova holatini boshqarish, zamonaviy web ilovalarida faqat foydalanuvchi tajribasini yaxshilash uchun emas, balki ularning xavfsizligini ta'minlash uchun ham juda muhimdir.Ushbu maqola, web dasturlashda ilova holatini boshqarish mexanizmlarini — cookie-fayllar va seanslarni chuqur tahlil qilishni maqsad qilgan. Bunda, ularning ishlash mexanizmi, xavfsizlikka bo‘lgan ta'siri, zamonaviy web ilovalarida qo‘llanilishi va ular bilan bog‘liq texnologiyalarni ko‘rib chiqamiz. Maqola PHP, ASP.NET, Java kabi dasturlash tillarida seans boshqaruvi bo‘yicha kod misollarini ham keltiradi.

Asosiy Qism

1. HTTP Protokolining Holatsizligi va Ilova Holatini Boshqarish Zaruriyati

Web dasturlashda asosiy protokol bo‘lgan HTTP protokoli holatsiz (stateless) hisoblanadi. Bu shuni anglatadiki, har bir so‘rov va javob mustaqil bo‘lib, ularning orasida uzluksiz bog‘lanish yoki holat saqlanmaydi. Demak, birinchi so‘rov amalga oshirilganida server foydalanuvchi haqida hech qanday ma'lumotga ega bo‘lmaydi. Har bir keyingi so‘rovda serverga yangi ma'lumotlar yuborilishi kerak bo‘ladi. Bunday holat, ayniqsa, foydalanuvchi sessiyasini davom ettirish, autentifikatsiya va boshqa jarayonlarni amalga oshirishda qiyinchiliklar tug‘diradi. Shuning uchun, ilova holatini boshqarish mexanizmlari, ya'ni **cookie-fayllar** va **seanslar** zarur bo‘ladi. Ushbu mexanizmlar, foydalanuvchi bilan uzluksiz aloqani ta'minlash, autentifikatsiya va xavfsizlikni boshqarish uchun muhim ahamiyatga ega.

2. Cookie-fayllar: Ishlash Mexanizmi va Qo‘llanilishi

Cookie-fayl — foydalanuvchining brauzerida saqlanadigan kichik hajmdagi ma'lumotlar fayli bo‘lib, u foydalanuvchi haqida ma'lumotlarni saqlashga mo‘ljallangan. Cookie-fayllar web saytlarning foydalanuvchi holatini saqlash,

ularning afzalliklari, autentifikatsiya ma'lumotlarini va boshqa maxsus ma'lumotlarni saqlash uchun ishlatiladi. Har bir cookie-fayl o‘zining nomi, qiymati, vaqt chegarasi va tegishli domeni bilan birga keladi. Cookie-fayllar, masalan, foydalanuvchi logini va parolini saqlash, foydalanuvchi afzalliklarini yodda tutish, saytga kiritilgan ma'lumotlarni saqlash uchun ishlatiladi.

Cookie-fayllarning eng muhim xususiyati shundaki, ular foydalanuvchi brauzeri tomonidan avtomatik ravishda yuboriladi va server tomonidan olinadi. Bu jarayon foydalanuvchi saytga qayta kirganida holatni saqlash imkoniyatini beradi. Cookie-fayllar bilan ishlashda quyidagi xavfsizlik masalalariga alohida e'tibor berish zarur:

- **Xavfsizlik (Secure) flag**: Cookie-fayl faqat HTTPS orqali uzatilishi kerak.
- **HttpOnly flag**: Cookie-fayl faqat server tomonidan o‘qilishi kerak, JavaScript orqali o‘qilmasligi zarur.
- **SameSite flag**: Cookie-fayl faqat o‘z domenida ishlatilishi kerak, bu esa XSS hujumlaridan himoya qiladi.

Misol (PHP):

```
php
КопироватьРедактировать
setcookie("user", "Xurshida", time() + 3600, "/"); // 1 soatlik cookie-fayl
```

Yuqoridagi kodda, "user" nomli cookie foydalanuvchining brauzerida saqlanadi va bir soat davomida amal qiladi.

3. Seanslar: Ishlash Mexanizmi va Xavfsizlik Masalalari

Seans (session) — serverda saqlanadigan ma'lumotlar to‘plamidir. Seanslar foydalanuvchiga bir nechta so‘rovlar davomida holatini saqlash imkoniyatini

beradi. Har bir foydalanuvchi uchun yagona seans identifikatori (ID) yaratiladi va bu identifikator cookie yoki URL orqali uzatiladi. Seanslar server tomonidan boshqariladi va foydalanuvchi so‘rovlari davomida ma'lumotlar serverda saqlanadi, bu esa foydalanuvchining shaxsiy ma'lumotlarini xavfsiz saqlash imkonini beradi.

Seanslar cookie-fayllardan farqli o‘laroq, serverda saqlanadi va foydalanuvchining brauzerida faqat seans identifikatori bo‘ladi. Bu, xavfsizlik nuqtai nazaridan, cookie-fayllarga nisbatan afzallikdir, chunki foydalanuvchi ma'lumotlari serverda himoyalangan holda saqlanadi.

Misol (PHP):

```
php
КопироватьРедактировать
session_start(); // Seansni ishga tushirish
$_SESSION["user"] = "Xurshida"; // Foydalanuvchi ma'lumotlarini saqlash
```

Bu kodda, foydalanuvchi nomi serverda saqlanadi va seans davomida foydalaniladi. Seans tugatilganda yoki ma'lum vaqt o‘tgach, ma'lumotlar o‘chiriladi.

4. Xavfsizlik va Web Dasturlashda Seans va Cookie Boshqaruvi

Web dasturlarda xavfsizlik masalalari eng dolzarb mavzulardan biridir. Cookie-fayllar va seanslar bilan ishlashda bir qator xavfsizlikka oid protokollarni qo‘llash zarur. Bu protokollar foydalanuvchi ma'lumotlarini xavfsiz saqlash, tasodifiy hujumlardan (CSRF, XSS) himoya qilish va autentifikatsiya jarayonlarini ishonchli amalga oshirish uchun muhimdir.

CSRF (Cross-Site Request Forgery) hujumlaridan himoya qilish uchun cookie-fayllarga **SameSite** flagini qo‘shish tavsiya etiladi. Bu flag cookie-faylning faqat o‘z saytida ishlatilishiga imkon beradi, bu esa tashqi saytlar orqali amalga oshirilgan so‘rovlarni bloklaydi.

XSS (Cross-Site Scripting) hujumlaridan himoya qilish uchun esa, cookie-fayllarni **HttpOnly** flagi bilan sozlash lozim. Bu flag cookie-faylni faqat server tomonidan o‘qilishi mumkin qilib, JavaScript orqali o‘qilishiga yo‘l qo‘ymaydi.

Misol (PHP) - SameSite flag:

php

КопироватьРедактировать

setcookie("user", "Xurshida", time() + 3600, "/", "", true, true); // Secure va HttpOnly flaglari qo‘shilgan

5. Seans va Cookie-fayllar Bilan Bog‘liq Zamonaviy Texnologiyalar

Bugungi kunda, seans va cookie-fayllar bilan ishlash uchun zamonaviy web ilovalarida bir qator innovatsion texnologiyalar qo‘llaniladi. Bular orasida **JWT (JSON Web Token)** va **OAuth 2.0** autentifikatsiya protokollari keng qo‘llanilmoqda. Ushbu protokollar foydalanuvchi autentifikatsiyasini xavfsiz tarzda amalga oshirishga yordam beradi va ilovalarda ko‘p bosqichli xavfsizlikni ta'minlaydi.

JWT (JSON Web Token) — bu foydalanuvchi autentifikatsiyasi uchun token asosidagi tizimdir, bu tizim foydalanuvchiga bir martalik tokenni berish orqali autentifikatsiya jarayonini boshqaradi. JWT tokenlari serverda saqlanmaydi, ularni foydalanuvchi o‘zi saqlaydi va keyingi so‘rovlarda tokenni serverga yuboradi.

Misol (JWT):

```
php
КопироватьРедактировать
$header = json_encode(["alg" => "HS256", "typ" => "JWT"]);
$payload = json_encode(["user" => "Xurshida"]);
$secret = "secretKey";
$jwt = base64_encode($header . "." . $payload . "." . hash_hmac('SHA256', $header . "." . $payload, $secret));
```

Xulosa

Web dasturlashda ilova holatini boshqarish, ayniqsa, cookie-fayllar va seanslar yordamida amalga oshiriladigan tizimlar zamonaviy web ilovalarining asosiy komponentlaridan biri bo‘lib, foydalanuvchi tajribasini yanada yuqori darajaga ko‘taradi. Cookie-fayllar va seanslar, o‘zlarining ishlash mexanizmlari orqali, foydalanuvchilarni autentifikatsiya qilish, ular orasidagi o‘zaro aloqalarni saqlash va xavfsizlikni ta'minlashda muhim rol o‘ynaydi. Cookie-fayllar foydalanuvchi brauzerida saqlanadigan kichik ma'lumotlar bo‘lib, ma'lum bir vaqt davomida foydalanuvchi holatini saqlash imkonini beradi, seanslar esa serverda ma'lumotlar saqlanadi, bu esa xavfsizlik nuqtai nazaridan afzallikdir.

Dastur ishlab chiqishda cookie-fayllar va seanslar yordamida foydalanuvchi ma'lumotlarini saqlash va boshqarish jarayonlari ko‘plab xavfsizlik protokollari bilan ta'minlanishi kerak. Bu xavfsizlik mexanizmlari, xususan, CSRF va XSS hujumlariga qarshi kurashish, shuningdek, cookie-fayllarda ma'lumotlar xavfsizligini ta'minlashda muhim ahamiyatga ega. Bunday tizimlarni to‘g‘ri boshqarish va ularning xavfsizligini oshirish orqali, zamonaviy web ilovalarining samaradorligini yanada oshirish mumkin.

Shuningdek, web dasturlarda cookie-fayllar va seanslar bilan ishlashda JWT (JSON Web Token) va OAuth 2.0 kabi zamonaviy texnologiyalarni qo‘llash, ilova xavfsizligini yanada mustahkamlashga imkon beradi. Ushbu texnologiyalar token

asosida autentifikatsiya qilish, foydalanuvchi ma'lumotlarini xavfsiz tarzda uzatish va saqlashda samarali yondashuvlarni taqdim etadi.

Web ilovalarida ilova holatini boshqarish mexanizmlarini to‘g‘ri qo‘llash, nafaqat foydalanuvchi tajribasini yaxshilash, balki ilovaning umumiy xavfsizligini oshirishda ham muhimdir. Kelajakda web dasturlashda cookie-fayllar va seanslar kabi mexanizmlarning o‘sishi va rivojlanishi, dasturchilarga yanada samarali va xavfsiz tizimlar yaratishga yordam beradi.

Foydalanilgan Adabiyotlar

1. **Eckel, R.** (2007). Java: How to Program. Pearson Education, Inc.
2. **Hewitt, L. & J. M.** (2010). Web Security for Developers. O'Reilly Media.
3. **Shafranov, D.** (2019). Dasturlash Asoslari. O'zbekiston Respublikasi Matbuot va Axborot Agentligi.
4. **Olsson, A.** (2014). Mastering OAuth 2.0. Packt Publishing.
5. **W3C.** (2020). Cookies and Privacy. World Wide Web Consortium (W3C).
6. **Lerman, E.** (2016). Web Security for Dummies. Wiley Publishing.
7. **Mozilla Developer Network (MDN).** (2021). HTTP Cookies.
8. **RFC 6265** (2011). HTTP State Management Mechanism. Internet Engineering Task Force (IETF).
9. **Peters, T.** (2012). PHP for Web Developers. Addison-Wesley Professional.
10. **Beal, V.** (2020). Session Management in Web Applications. Webopedia.
11. **Sommerville, I.** (2011). Software Engineering. Addison-Wesley.
12. **Snyder, D. & Stevens, L.** (2013). Professional Web APIs: The Web 2.0 Developer's Guide. Wiley.

13. **Zeldman, J.** (2010). Designing with Web Standards. New Riders.

14. **Kaufman, C.** (2017). Web Application Security: Exploitation and Countermeasures for Modern Web Attacks. Wiley.

15. **Haverbeke, M.** (2018). Eloquent JavaScript. No Starch Press.