

Федеральное государственное бюджетное образовательное учреждение высшего  
образования

Казанский государственный энергетический университет  
Кафедра информационных технологий и интеллектуальных систем

## Задачи на Cookie

Методические указания для выполнения лабораторных работ



Казань, 2024

## Лабораторная работа. Использование Cookies

**Цель работы:** формирование умений использования Cookies.

### Теоретические сведения

#### Cookies

Cookie является решением одной из наследственных проблем HTTP спецификации. Эта проблема заключается в непостоянстве соединения между клиентом и сервером, как при FTP или Telnet сессии, т.е. для каждого документа (или файла) при передаче по HTTP протоколу посылается отдельный запрос. Включение cookie в HTTP протокол дало частичное решение этой проблемы и не только этой.

Cookie это небольшая порция информации, которую сервер передает клиенту. Клиент (браузер) будет хранить эту информацию, и передавать ее серверу с каждым запросом как часть HTTP заголовка. Некоторые cookie хранятся только в течение одной сессии, они удаляются после закрытия браузера. Другие, установленные на некоторый период времени, записываются в файл. Эти файлы вы можете видеть у себя на компьютере в каталоге \Documents and Settings\Имя\_пользователя\Cookies\.

Сами по себе cookies не могут делать ничего, это только лишь некоторая информация. Однако, сервер может реагировать на содержащуюся в cookies информацию. Например, в случае авторизованного доступа к чему-либо через WWW, в cookies можно сохраняется login и password в течение сессии, что позволяет не вводить их при запросе каждого запрошенного документа. Другой пример: cookies могут использоваться для построения персонализированных страниц. Чаще всего встречается такое - на некотором сервере вас просят ввести свое имя, и каждый раз, когда вы заходите на первую страницу этого сервера, вам пишут что-то типа "Здравствуй, Вася Пупкин!". На использовании cookies также часто строят функцию оформления заказа в онлайн-магазинах.

Cookie является частью HTTP заголовка. Полное описание поля Set-Cookie HTTP заголовка:

```
Set-Cookie: NAME=value; EXPIRES=date; DOMAIN=domain_name;
PATH=path; SECURE
```

Минимальное описание поля Set-Cookie HTTP заголовка:

```
Set-Cookie: NAME=VALUE;
```

**NAME=VALUE** - NAME - переменная (имя cookie), VALUE - значение.

**expires=DATE** - время хранения cookie (формат записи "expires=Monday, DD-Mon-YYYY HH:MM:SS GMT").

Пример:

```
expires=Friday,31-Dec-99 23:59:59 GMT;
```

Если этот атрибут не указан, то cookie хранится в течение одного сеанса, до закрытия браузера.

**domain=DOMAIN\_NAME** - домен, для которого значение cookie действительно. Например, "domain=ipm.kstu.ru".

Если этот атрибут опущен, то по умолчанию используется доменное имя сервера, на котором было задано значение cookie.

**path=PATH** - этот атрибут устанавливает корневой путь к документам, для которых действительно значение cookie. Например, указание "path=/students/pupkin/" приведет к тому, что значение cookie будет действительно для всех документов Васи Пупкина в директории /students/pupkin/. Для того, чтобы cookie отсылались при запросе любого документа сервера, необходимо указать корневой каталог сервера ("path=/").

Если этот атрибут не указан, то значение cookie распространяется только на документы в той же директории, что и документ, в котором было установлено значение cookie.

**secure** - если стоит этот маркер, то информация cookie пересылается только через HTTPS (HTTP с использованием SSL - Secure Socket Level), в зашифрованном

виде. Если этот маркер не указан, то информация пересылается обычным способом.

### **Установка cookie с помощью HTML.**

Простейший способ выставить cookie - использовать соответствующий META-тег в заголовке <HEAD> </HEAD> любого HTML документа. Это выглядит следующим образом:

```
<META HTTP-EQUIV="Set-Cookie" CONTENT="NAME=value; EXPIRES=date;
DOMAIN=domain_name; PATH=path; SECURE">
```

С помощью HTML можно устанавливать несколько cookie разом:

```
<META HTTP-EQUIV="Set-Cookie" CONTENT="NAME1=value1;
EXPIRES=date1; DOMAIN=domain_name1; PATH=path1; SECURE">
<META HTTP-EQUIV="Set-Cookie" CONTENT="NAME2=value2;
EXPIRES=date2; DOMAIN=domain_name2; PATH=path2; SECURE">
```

### **Синтаксис HTTP заголовка для поля Cookie при запросе документа с сервера.**

Когда запрашивается документ с HTTP сервера, браузер проверяет свои cookie на предмет соответствия домену сервера (DOMAIN), пути к документу (PATH), если время хранения истекло, то он удаляется. В случае, если найдены удовлетворяющие всем условиям значения cookie, браузер посылает их серверу в виде пары имя/значение:

```
Cookie: NAME1=STRING1; NAME2=STRING2 ...
```

### **Дополнительные сведения**

В случае, если cookie принимает новое значение при имеющемся уже в браузере cookie с совпадающими параметрами NAME, domain и path, то старое значение заменяется новым. В остальных случаях новые значения cookie добавляются к старым.

Использование expires не гарантирует сохранность cookie в течение заданного периода времени, поскольку клиент (браузер) может удалить запись из-за нехватки выделенного места или каких-либо других причин.

Клиент (браузер) может иметь ограничения для cookies, например:

- всего может храниться до 300 значений cookies
- каждый cookie не может превышать 4Кбайт
- с одного сервера или домена может храниться до 20 значений cookie

Если ограничение 300 или 20 превышает, то удаляется первая по времени запись. При превышении лимита объема в 4Кбайт корректность значения cookie страдает - отрезается кусок записи (с начала этой записи) равный превышению объема.

### **Получение и использование cookie сервером с помощью SSI.**

При наличии SSI на сервере, директивой `<!--#echo var="..."-->` можно считать любые переменные окружения, в том числе и ранее заданные значения cookie (переменная HTTP\_COOKIE).

Примеры:

```
<!--#echo var="HTTP_COOKIE"-->
```

Вставка всех полученных cookies в документ.

```
<!--#if expr="$HTTP_COOKIE=/user\=old/" -->
```

проверяем, содержится ли в cookies "user=old".

```
<META HTTP-EQUIV="Set-Cookie" CONTENT="<!--#echo  
var="QUERY_STRING"-->";">
```

вставляем строку запроса, посланную методом GET для установки cookie.

## **Практические задания**

### **Задание 1**

1. Сделайте, чтобы ваша первая страница узнавала посетителя (приветствовала что "вы снова зашли на сайт студента "Ваше ФИО", если первый раз на странице, то писала "я вас не знаю" "). Примените директиву if из SSI (посмотрите пояснение к лабораторной работе 6).

2. Подробнее, что должно происходить:

При первом открывании страницы, она должна послать клиенту cookie.

При последующих запросах страницы, клиент должен посылать cookie серверу.

Сервер, получив это cookie, должен выдавать приветствие, в противном случае сообщать, что вас не знает.

3. Проверьте работоспособность на сервере.

4. Укажите, в виде комментариев, для чего предназначена каждая SSI команда.

## **Задание 2**

1. Создайте страницу с помощью которой можно "зарегистрировать" себя:

- новичком,

- средним пользователем,

- опытным пользователем.

Оформите регистрацию применив формы (radio или select). Данные должны отправляться на вторую страницу методом GET.

2. На второй странице сделайте вставку переменной и её значения в тег <META> для установки cookie.

3. Сделайте, на этой странице, обработку посылаемых cookies с помощью директивы IF (из SSI) так, что бы страница выдавала, кто зашел на сайт (новичок, средний или опытный пользователь).

4. Проверьте работоспособность на сервере.

5. Укажите, в виде комментариев, для чего предназначена каждая SSI команда.

## **Задание 3**

1. Создайте страницу с помощью, которой можно "обнулить" ваши cookies, т.е. при запросе этой страницы передавались ваши cookies с просроченной датой.

2. Ссылку на эту страницу оформите с помощью кнопки "submit" применив формы.

3. Проверьте работоспособность на сервере.

*К сдаче лабораторной предоставляются: работающие страницы на сервере.*