

HTTP-заголовки, часть 2

JSP & Servlets
9 уровень, 6 лекция

ОТКРЫТА

7.1 Заголовок Keep-Alive

И еще несколько полезных заголовков. Заголовок **Keep-Alive** указывает серверу, что соединение нужно оставить открытым: сервер не будет закрывать соединение сразу после отправки ответа. Это приведет к тому, что следующий запрос от этого же клиента к серверу будет выполнен быстрее.

Пример такого заголовка:

```
Connection: Keep-Alive
```

Но если все клиенты будут требовать постоянного соединения, тогда проблемы начнутся уже у сервера. Сервер будет или недоступен, или начнет закрывать соединения по своему выбору.

7.2 Заголовок Cache-Control

С помощью заголовка **Cache-Control** можно управлять кэшированием контента. Грамотно настроенное кэширование ускоряет работу с контентом, криво настроенное создает проблемы на ровном месте.

Чтобы **отключить кэширование** нужно написать такой заголовок:

```
Cache-Control: no-cache, no-store, must-revalidate
```

В кэше не должно сохраняться ничего — ни по запросам клиента, ни по ответам сервера. Запрос всегда отправляется на сервер, ответ всегда загружается полностью.

Также можно включить **самый примитивный и надежный тип кэширования**:

```
Cache-Control: no-cache
```

Перед тем, как выдать копию, кэш запрашивает исходный сервер на предмет актуальности ресурса.

Можно указать **время кэширования ресурса в секундах**. Выглядеть такой заголовок будет так:

```
Cache-Control: max-age=31536000
```

Этот заголовок задает максимальное время хранения контента в кэше.

Более детально про кеширование ты можешь [почитать тут](#)

7.3 Cookie

Сервер может хранить данные на стороне клиента. Такие данные называются **cookie**. Впрочем, cookie может сохранить и клиент. Они бывают очень полезны обеим сторонам.

Например, ты заходишь на сайт, а ты на нем уже авторизирован. То есть, когда ты залогинился в него в прошлый раз, сервер приказал браузеру сохранить у себя информацию об успешном логине определенного пользователя.

Вот как выглядит Cookie в запросе:

Cookie: name=value;name2=value2;nameN=valueN00

Cookie обычно хранит браузер и **они привязаны к определенному домену**. Когда ты снова заходишь на тот же домен, cookie автоматически добавляются к http-запросу и http-ответу. Сервер/домен не может получить cookie, которые хранит в браузере другой сервер/домен.

У каждого cookie есть 4 основных параметра:

- **имя;**
- **значение;**
- **время действия (сколько времени их хранить);**
- **домен, к которому привязаны cookie.**

Cookie хранятся и передаются в текстовом виде, так что и имя, и значение – это строки. Если время действия cookie не указано, то они уничтожаются после закрытия браузера.

7.4 Session

После того, как пользователь залогинился на сайте, говорят, что между сайтом и сервером установилась сессия.

Сервер у себя создает специальный объект – `HttpSession`, где хранит всю нужную информацию для работы с авторизованным клиентом. А уникальный номер этого объекта хранит в браузере в виде Cookie.

Веб-сервера на Java обычно используют имя `JSESSIONID` для хранения идентификатора сессии. Выглядит это примерно так:

Cookie: JSESSIONID =ABAD1D

На стороне сервера можно задать время существования сессии, а также то, будет ли она автоматически закрываться при закрытии браузера.

[< Предыдущая лекция](#)

[Следующая лекция >](#)

+13

Комментарии

популярные

новые

старые

JavaCoder

Введите текст комментария



У ЭТОЙ СТРАНИЦЫ ЕЩЕ НЕТ НИ ОДНОГО КОММЕНТАРИЯ

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский 



