# Работа сервлета с сессиями

JSP & Servlets 12 уровень, 3 лекция

ОТКРЫТА

# 4.1 Знакомство с HttpSession

Если несколько запросов идут от одного клиента, то говорят, что между клиентом и сервером установилась сессия. Для контроля этого процесса у контейнера есть специальный объект HttpSession.

Когда клиент обращается к сервлету, то контейнер сервлетов проверяет, есть ли в запросе параметр ID сессии. Если такой параметр отсутствует (например, клиент первый раз обращается к серверу), тогда контейнер сервлетов создает новый объект HttpSession, а также присваивает ему уникальный ID.

Объект сессии сохраняется на сервере, а ID отправляется в ответе клиенту и по умолчанию сохраняется на клиенте в куках. Затем, когда приходит новый запрос от того же клиента, то контейнер сервлетов достает из него ID, и по этому ID находит правильный объект HttpSession на сервере.

Получить объект сессии можно из запроса (объект HttpServletRequest), у которого нужно вызвать метод getSession(). Он возвращает объект HttpSession.

Зачем нужна сессия? В ней можно хранить информацию о клиенте между вызовами. У нее внутри есть что-то вроде HashMap, в котором можно хранить объекты по ключам. И несколько методов для этого:

	Методы	Описание
1	setAttribute(String name, Object o)	Добавляет объект внутрь сессии
2	<pre>getAttribute(String name)</pre>	Получает объект из сессии
3	removeAttribute(String name)	Удаляет объект из сессии

Давай напишем сервлет, который будет суммировать все передаваемые ему числа из разных запросов:

```
1
     public class CalculatorServlet extends HttpServlet {
 2
         @Override
 3
         protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
               // Получаем атрибут "sum" из сессии
 4
 5
             HttpSession session = request.getSession();
             Integer sum = (Integer) session.getAttribute("sum");
 6
 7
             //Обрабатываем ситуацию, когда такого атрибута у сессии еще нет
 8
             if (sum == null)
9
                  sum = 0;
10
              // Получаем параметр "n" из запроса
11
             String n = request.getParameter("n");
12
             sum += Integer.parseInt(n);
13
14
              // Записываем атрибут "sum" в сессию
15
             session.setAttribute("sum", sum);
16
17
             // Печатаем HTML в качестве ответа для браузера
18
             PrintWriter out = response.getWriter();
19
```

```
out.println("<html>");
20
              out.println("<head> <title> CalculatorServlet </title> </head>");
21
22
              out.println("<body>");
              out.println("<h1> Sum == " + sum + "</h1>");
23
              out.println("</body>");
24
              out.println("</html>");
25
         }
26
27
     }
```

# 4.2 Подробнее об HttpSession

Что еще важного мы не сказали об объекте HttpSession?

Bo-первых, это имя JSESSIONID. Именно под ним в куках хранится ID сессии. Как видите, запомнить его довольно легко: J+SESSION+ID.

Во-вторых, у сессии есть еще несколько полезных методов:

	Методы	Описание
1	<pre>getAttributeNames()</pre>	Возвращает список всех ключей, которые храниться в сессии
2	<pre>getId()</pre>	Возвращает ID-сессии (строка)
3	isNew()	Возвращает true, если объект сессии был создан в текущем запросе
4	setMaxInactiveInterval(int seconds)	Устанавливает интервал неактивности сессии в секундах
5	<pre>invalidate()</pre>	Удаляет из сессии все объекты

Тут все методы очевидны, а про setMaxInactiveInterval() мы поговорим немного подробнее.

Если сервер будет хранить десятки тысяч сессий, включая данные клиентов, которые заходили к нему в прошлом месяце, то у него банально закончиться память. Поэтому есть способ установить "время жизни сессии".

Если в течение interval времени сессией никто не пользовался, то она самоочищается — из нее удаляются все объекты, которые она хранила. Это сделано для экономии памяти.

По умолчанию этот интервал равен 1800 секундам == 30 минутам. Если установить значение -1, то сессия будет "вечной" и удалится только когда пользователь закроет вкладку браузера (ну или клиент разорвет соединение).

### Примеры:

```
// получение всех ключей

Enumeration keys = session.getAttributeNames();

while( keys.hasMoreElements() ){

System.out.println( (String) keys.nextElement() );

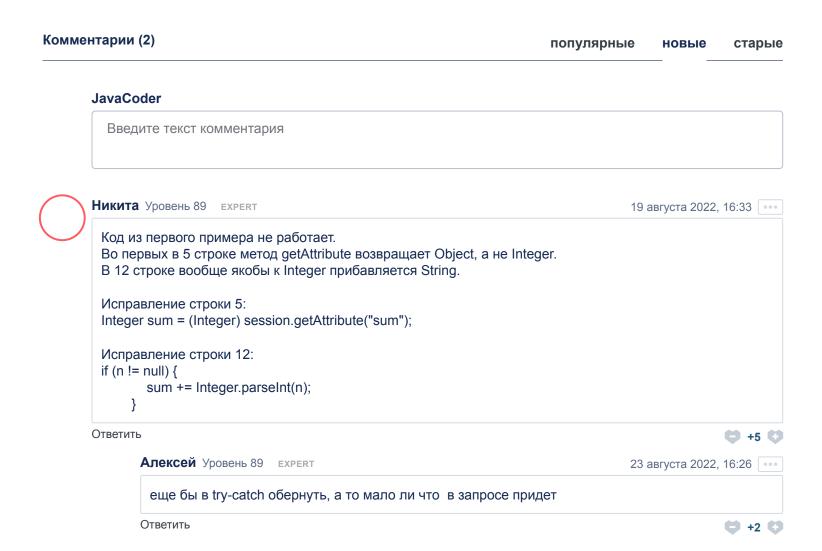
}
```

```
// установка интервала неактивности
session.setMaxInactiveInterval(60*60*24); // 1 день
session.setMaxInactiveInterval(-1); // до закрытия браузера
```

```
1 // удаление всех данных из сессии
2 session.invalidate();
```



КОМПАНИЯ



Курсы программирования	Пользователи	О нас
Kypc Java	Статьи	Контакты
Помощь по задачам	Форум	Отзывы
Подписки	Чат	FAQ
Задачи-игры	Истории успеха	Поддержка
	Активности	

СООБЩЕСТВО



ОБУЧЕНИЕ

RUSH

статьи об эффективном обучении и карьере Java-девелопера.

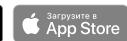
## ПОДПИСЫВАЙТЕСЬ

#### ЯЗЫК ИНТЕРФЕЙСА

Русский

#### СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ







"Программистами не рождаются" © 2023 JavaRush