

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	getAttribute(String name)	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 {
4         // Получаем атрибут “sum” из сессии
5         HttpSession session = request.getSession();
6         Integer sum = (Integer) session.getAttribute("sum");
7         //Обрабатываем ситуацию, когда такого атрибута у сессии еще нет
8         if (sum == null)
9             sum = 0;
10
11         // Получаем параметр “n” из запроса
12         String n = request.getParameter("n");
13         sum += Integer.parseInt(n);
14
15         // Записываем атрибут “sum” в сессию
16         session.setAttribute("sum", sum);
17
18         //Печатаем HTML в качестве ответа для браузера
19         PrintWriter out = response.getWriter();
```

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

4.2 Подробнее об HttpSession

Что еще важного мы не сказали об объекте HttpSession?

Во-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	<code>getAttributeNames()</code>	Возвращает список всех ключей, которые храниться в сессии
2	<code>getId()</code>	Возвращает ID-сессии (строка)
3	<code>isNew()</code>	Возвращает true, если объект сессии был создан в текущем запросе
4	<code>setMaxInactiveInterval(int seconds)</code>	Устанавливает интервал неактивности сессии в секундах
5	<code>invalidate()</code>	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:

```
1 // получение всех ключей
2 Enumeration keys = session.getAttributeNames();
3 while( keys.hasMoreElements() ){
4     System.out.println( (String) keys.nextElement() );
5 }
```

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

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

-

+11

+

Комментарии (2)

популярные

новые

старые

JavaCoder

Введите текст комментария



Никита

Уровень 89

EXPERT

19 августа 2022, 16:33

...

Код из первого примера не работает.
Во первых в 5 строке метод `getAttribute` возвращает `Object`, а не `Integer`.
В 12 строке вообще якобы к `Integer` прибавляется `String`.

Исправление строки 5:
`Integer sum = (Integer) session.getAttribute("sum");`

Исправление строки 12:

```
if (n != null) {  
    sum += Integer.parseInt(n);  
}
```

Ответить

-

+5

+

Алексей

Уровень 89

EXPERT

23 августа 2022, 16:26

...

еще бы в `try-catch` обернуть, а то мало ли что в запросе придет

Ответить

-

+2

+

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

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский 

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

