

Формирование ответов сервлета

JSP & Servlets
12 уровень, 2 лекция

ОТКРЫТА

3.1 Класс HttpServletResponse

После вызова метода `service()` твоего сервлета веб-контейнер сформирует ответ клиенту на основе объекта `HttpServletResponse`. Так что если хочешь как-то повлиять на ответ клиенту, то стоит модифицировать этот объект.

Ответ сервера состоит из трех частей:

- Строка статуса (например: 200 OK)
- Заголовки (headers)
- Тело ответа (response body)

И методы класса `HttpServletResponse` тоже делятся на 3 группы? Ну почти:

	Методы	Описание
1	<code>void setStatus(int sc)</code>	Устанавливает код статуса ответа.
2	<code>void sendError(int sc)</code>	Отправляет клиенту ошибку с указанным кодом.
3	<code>void sendError(int sc, String msg)</code>	Отправляет клиенту ошибку с указанным кодом и сообщением.
4	<code>void addHeader(String name, String value)</code>	Добавляет заголовок в список заголовков ответа.
5	<code>void setHeader(String name, String value)</code>	Меняет заголовок в списке заголовков ответа.
6	<code>boolean containsHeader(String name)</code>	Проверяет, есть ли уже такой заголовок.
7	<code>void addCookie(Cookie cookie)</code>	Добавляет Cookie к ответу.
8	<code>void sendRedirect(String location)</code>	Перенаправляет клиента на другой URL.
9	<code>String encodeRedirectURL(String url)</code>	Кодирует указанный URL для его использования в методе <code>sendRedirect</code> .
10	<code>String encodeURL(String url)</code>	Кодирует указанный URL, включая в него идентификатор сессии.
11	<code>void setContentType(String type)</code>	Устанавливает MIME-тип результата.
12	<code>void setContentLength(int len)</code>	Устанавливает длину тела ответа.
13	<code>void setCharacterEncoding(String charset)</code>	Устанавливает набор кодировок ответа.
14	<code>void setBufferSize(int size)</code>	Устанавливает размер буфера для тела ответа.

15	<code>boolean isCommitted()</code>	Проверяет, записан ли уже буфер в ответ.
16	<code>void flushBuffer()</code>	Записывает содержимое буфера в ответ.
17	<code>void reset()</code>	Сбрасывает все данные, которые хранятся в буфере, заголовки и коды ответа.
18	<code>void resetBuffer()</code>	Очищает буфер ответа.

Методы тут достаточно тривиальны. Но расскажу ниже несколько нюансов.

Буфер. Ответ твоего сервлета пишется в буфер, а не сразу отправляется пользователю. Поэтому на каком-то этапе (если, например произошла ошибка), ты можешь сбросить (стереть) все, что записано в буфер. Можно даже вызвать метод `reset()` и стереть не только содержимое буфера, но и заголовки с кодом ответа.

3.2 redirect()

Второй важный момент — это редирект. Если твой сервлет примет решение перенаправить клиента на другой URL, то тебе нужно как-то этот URL отправить клиенту. Это можно сделать с помощью метода `sendRedirect`.

Но есть важный нюанс. URI может содержать более широкий диапазон символов, чем разрешено в response body. Поэтому URL перед вызовом метода `sendRedirect()` нужно сначала перекодировать в допустимый набор символов. Для этого и есть специальный метод `encodeRedirectURL(String url)`. Воспользуйся им.

Пример редиректа:

```
1 public class RedirectServlet extends HttpServlet {
2     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Exception {
3         String newUrl = "http://google.com?q=Учись гуглить!";
4         String redirectUrl = response.encodeRedirectURL(newUrl);
5         response.sendRedirect(redirectUrl);
6     }
7 }
```

3.3 Метод getWriter()

А теперь мы научимся писать свой текст в качестве response body. Для этого у класса `HttpServletResponse` есть специальный метод `getWriter()`, который возвращает объект `PrintStream`. Если кто забыл, именно такой тип имеет поле с именем `System.out`.

Для того, чтобы написать какой-то текст в качестве ответа сервлета, нужно:

- Получить объект `PrintStream`, вызвав метод `response.getWriter()`.
- Записать в объект `PrintStream` все, что считаешь нужным (все данные запишутся в буфер).
- Отправить буфер пользователю, вызвав метод `close()` у `PrintStream`.

Давай напишем сервлет, который складывает два числа `a` и `b` и возвращает пользователю результат:

```
1 public class CalculatorServlet extends HttpServlet {
2     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Exception {
3         // Получаем параметр “a” и “b” из запроса
4         String a = request.getParameter("a");
5         String b = request.getParameter("b");
6
7         try {
8             // Преобразовываем строки в числа и считаем сумму
9             int sum = Integer.parseInt(a) + Integer.parseInt(b);
10        }
```

```
11 // Печатаем HTML в качестве ответа для браузера
12 response.setContentType("text/html;charset=UTF-8");
13 PrintWriter out = response.getWriter();
14
15 out.println("<html>");
16 out.println("<head> <title> CalculatorServlet </title> </head>");
17 out.println("<body>");
18 out.println("<h1> Sum == " + sum + "</h1>");
19 out.println("</body>");
20 out.println("</html>");
21 } finally {
22     out.close();
23 }
24 }
25 }
26
27
```

[← Предыдущая лекция](#)

[Следующая лекция →](#)

− +12 +

Комментарии (5)

популярные

новые

старые

JavaCoder

Введите текст комментария

kv0ut

Уровень 51

4 января, 18:54

...

В приведенном коде переменная out имеет тип PrintWriter, метод getWriter возвращает объект Print Stream. Данный код не скомпилируется.

Ответить

−

0

+

Артём

Инженер по автоматизации в YOKOGAWA

23 ноября 2022, 17:13

...

Класс RedirectServlet перекидывает на страницу Google, но вместо русских символов знаки ?. Как победить такую проблему?

Нашел ответ, вдруг кому понадобится:
String query = "q=" + URLEncoder.encode("Учись гуглить!", "UTF-8");
String url = "http://google.com?" + query;
String redirectUrl = response.encodeRedirectURL(url);

Ответить

−

0

+

Anonymous #3145256

Уровень 3

23 августа 2022, 14:00

...

В последнем примере 13 строку нужно вынести из блока try, потому что в finally объект out недоступен, область видимости и всё такое

Ответить

−

+4

+

Александр Ц.

System Engineer

EXPERT

5 октября 2022, 16:14

...

... и внести в try-with-resources, избавившись от finally

Ответить

−

+1

+

Антон

Уровень 90

EXPERT

6 октября 2022, 17:27

...

Скорее всего, finally и close() тут указали в учебных целях, чтобы использование метода было наглядно.
В try-with-resources этот момент не так очевиден.

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 

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

