

Обслуживание запросов в сервлете

JSP & Servlets
12 уровень, 1 лекция

ОТКРЫТА

2.1 Класс HttpServletRequest

Большую часть работы твой сервлет должен сделать на основе информации, полученной из запроса. За него отвечает объект `HttpServletRequest`, который контейнер передаст в твой сервлет (в метод `service` или методы `doGet()`, `doPost()` и т. д.)

У этого объекта довольно много методов, так как он просто хранит данные запроса, а через него можно взаимодействовать с контейнером.

Методы можно разделить на 2 большие группы:

- Методы, связанные с авторизацией пользователя
- Методы по работе с данными запроса

Методы по авторизации пользователя я приведу в виде таблицы, но разбирать их мы не будем. Дело в том, что их очень редко используют для авторизации пользователя. Все популярные фреймворки используют свои, гораздо более продвинутые подходы к авторизации.

Я должен привести их список, но опять-таки, я не видел, чтобы кто-то их использовал.

	Метод	Описание
1	<code>authenticate(HttpServletResponse)</code>	Выполняет аутентификацию response’a
2	<code>changeSessionId()</code>	Принудительно меняет ID сессии
3	<code>getAuthType()</code>	Возвращает тип аутентификации, который используется: BASIC_AUTH, FORM_AUTH, CLIENT_CERT_AUTH, DIGEST_AUTH
4	<code>getRemoteUser()</code>	Возвращает login пользователя
5	<code>getRequestedSessionId()</code>	Возвращает SessionID клиента
6	<code>getSession()</code>	Возвращает объект HttpSession
7	<code>getUserPrincipal()</code>	Возвращает объект java.security.Principal
8	<code>login(username, password)</code>	Выполняет логин пользователя
9	<code>logout()</code>	Разлогинивает сессию пользователя

2.2 Данные запроса

Вторая группа методов гораздо интереснее. Что за данные у нас есть в запросе?

- http-метод
- URI
- Параметры
- Заголовки

Посмотрим, какие методы есть для работы с ними:

	Метод	Описание
1	<code>getMethod()</code>	Возвращает HTTP-метод: GET, POST, DELETE, ...
2	<code>getRequestURI()</code>	Возвращает URI запроса: http://javarush.ru/my/data
3	<code>getRequestURL()</code>	Возвращает URL запроса: http://javarush.ru/my/data
4	<code>getQueryString()</code>	Возвращает Query, то есть все, что идет после знака ?
5	<code>getParameterMap()</code>	Возвращает список параметров запроса
6	<code>getParameter(String name)</code>	Возвращает значение параметра по его имени
7	<code>getContentType()</code>	Возвращает MimeType request body
8	<code>getReader()</code>	Reader, чтобы прочитать request body как текст
9	<code>getInputStream()</code>	InputStream, чтобы прочитать request body как byte[]
10	<code>getSession()</code>	Возвращает объект HttpSession
11	<code>getCookies()</code>	Возвращает массив объектов Cookie[]
12	<code>getHeaderNames()</code>	Возвращает список заголовков, только имена
13	<code>getHeader(String name)</code>	Возвращает значение заголовка по имени
14	<code>getServletPath()</code>	Возвращает часть URL, которая относится к сервлету
15	<code>getContextPath()</code>	Возвращает часть URI, которая задает контента запроса

И это даже не все методы...

Хотя после того, как мы изучили протокол HTTP и научились работать с HttpClient’ом, тут уже все более-менее знакомо, не так ли?

Давай напишем сервлет, в который можно передать текст и цвет, а он будет возвращать HTML-страницу с этим текстом, написанным указанным цветом. Как тебе идея?

Начнем с написания своего сервлета:

```
1 public class ColorTextServlet extends HttpServlet {
2     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Exception {
3         //напишите тут ваш код
4     }
5 }
```

Теперь нам нужно получить текст и цвет, переданные у URI пользователем:

```
1 public class ColorTextServlet extends HttpServlet {
2     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Exception {
3
4         // Получаем параметр “text” и “color” из запроса
5         String text= request.getParameter("text");
6         String color = request.getParameter("color");
7
8     }
9 }
```

И наконец нужно вывести текст в виде HTML. Мы разберем это в следующей лекции, но тут я немного подсказу:

```
1 public class ColorTextServlet extends HttpServlet {
2     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Exception {
3
4         // Получаем параметр “text” и “color” из запроса
5         String text = request.getParameter("text");
6         String color = request.getParameter("color");
7
8
9         // Печатаем HTML в качестве ответа для браузера
10        response.setContentType("text/html;charset=UTF-8");
11        PrintWriter out = response.getWriter();
12        try {
13            out.println("<html>");
14            out.println("<head> <title> ColorTextServlet </title> </head>");
15            out.println("<body>");
16            out.println("<h1 style='color:"+color+"'>"+text+"</h1>");
17            out.println("</body>");
18            out.println("</html>");
19        } finally {
20            out.close();
21        }
22    }
23 }
```

[< Предыдущая лекция](#)

[Следующая лекция >](#)

+11

Комментарии (4)

популярные новые старые

JavaCoder

Введите текст комментария

Сергей Филиппов Уровень 89 EXPERT

27 сентября 2022, 11:54

Запятая лишняя, вот так работает:
out.println("<h1 style=color:" + color + ">" + text + "</h1>");

Ответить 0

Vladislav Shamshurin Уровень 1

19 августа 2022, 11:44

getRequestURI() - Возвращает URI запроса: http://javarush.ru/my/data
Должен возвращаться /my/data

Ответить +2

Никита Уровень 89 EXPERT

19 августа 2022, 14:25

Не уверен, скорее в url должен возвращаться просто http://javarush.ru

Ответить +2

Сергей Уровень 79

26 октября 2022, 19:46

Method: GET
requestURI: /hello-servlet

requestURL: http://localhost:8080/hello-servlet
queryString: name=Maxim&userId=123
parameter: Maxim

Из такого url: http://localhost:8080/hello-servlet?name=Maxim&userId=123

Ответить

👍

+3

👍

👍

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

