

JSP: декларативный способ задания сервлета

JSP & Servlets
13 уровень, 0 лекция

ОТКРЫТА

1.1 Знакомство с JSP

Есть два популярных способа написания сервлета: **императивный** и **декларативный**. С первым мы уже разобрались — это, собственно, Servlet. Второй называется JSP (Java Server Pages), с ним мы сейчас и познакомимся.

Пример JSP сервлета:

```
1 <html>
2   <body>
3       <% out.print(2*5); %>
4   </body>
5 </html>
```

Не очень-то похоже на привычный нам классический Servlet, не правда ли? Так и есть. **JSP – это HTML-страница со вставками Java-кода (выделено зеленым).**

Видишь ли, **если в сервлете у тебя много Java-кода и мало HTML-кода, тогда тебе удобнее применять классический сервлет**. А что делать, если тебе нужна большая HTML-страница, в которой всего пара строк меняется сервером?

В этом случае проще всего было бы создать эту HTML-страницу и каким-то образом выполнить Java-код на сервере прямо в ней.

1.2 Компиляция JSP

Давай рассмотрим еще один пример:

```
1 <html>
2   <body>
3       <%
4           double num = Math.random();
5           if (num > 0.95) {
6               %>
7               <h2>Ты счастливчик, парень!</h2><p>(<%= num %>)</p>
8               <%
9               }
10              %>
11   </body>
12 </html>
```

Мы получаем случайное число, и если оно больше 0.95, то выводим текст “Ты счастливчик, парень!”

Синим цветом тут выделен Java-код. Обычным (не выделен) – HTML-код, **а красным — служебные теги**, которые помогают понять, где Java-код, а где HTML.

Замечаешь некую странность? Закрывающая фигурная скобка **“}”** находится в другом **“блоке кода”**. Как правильно писать такой код? Как это вообще работает?

Ответ будет супер простым :)

Веб-сервер после того, как найдет у себя JSP-файл, скомпилирует его в классический сервлет. На основе приведенной выше JSP-страницы будет сгенерирован такой Servlet:

```
1 public class HelloServlet extends HttpServlet {
2     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Exception {
3         PrintWriter out = response.getWriter();
4         out.print("<html> ");
5         out.print("<body>");
6         double num = Math.random();
7         if (num > 0.95) {
8             out.print("<h2>Ты счастливчик, парень! </h2><p>(" + num + ")</p>");
9         }
10        out.print("</body>");
11        out.print("</html>");
12    }
13 }
```

Веб-контейнер просто сгенерировал код сервлета, где HTML превратился в текст, а вставки Java-кода стали обычным Java-кодом!

< Предыдущая лекция

Следующая лекция >

− +10 +

Комментарии (2)

популярные

новые

старые

JavaCoder

Введите текст комментария

Alexander

Работает в Сканпорт

EXPERT

21 декабря 2022, 16:53

<h2>Ты счастливчик, парень!</h2>
(<%= num %>)

Девушки: ну-да, ну-да мы же чмо, нас и проигнорить можно...

Ответить

−

0

+

Dinis Shayslamov

Уровень 78

EXPERT

7 октября 2022, 23:26

в третьей строке переменная точно resp а не response?

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

