

Знакомство с сервлетами

JSP & Servlets
12 уровень, 0 лекция

ОТКРЫТА

1.1 interface Servlet

Сегодня начинаем новую и интересную тему — **сервлеты**. Именно добавление сервлетов в Java привело к тому, что Java сейчас — де-факто стандарт для больших серверных приложений. 80% всего корпоративного ПО в мире пишется на Java. А в Китае так и все 100%. Так что же такое сервлеты?

Сервлет — это именно то, что превращает Java-программу в веб-сервис и позволяет ей обрабатывать запросы от клиентов. А дело было так...

В 90-е годы, сразу после появления всемирной паутины, появились веб-клиенты (браузеры) и веб-сервера. **Веб-сервера** обычно просто раздавали через интернет файл, которые у них хранились: html-страницы, скрипты, картинки и т. п.

В какой-то момент все пришли к выводу, что нужно бы сделать обе стороны поумнее. В HTML-страницы добавили JavaScript, а серверам добавили плагины — **специальные скрипты, которые вызывались в ответ на определенные запросы** и позволяли сделать поведение сервера гибче и умнее.

Так вот **сервлет** — это такой **Java-плагин**, который встраивался в **Java веб-сервер** и позволял ему выполнять Java-код при запросе на определенные страницы. А уже этот Java-код, представленный классом, унаследованным от класса Servlet, делал то, что задумали его разработчики.

И как ты уже знаешь, самый популярный Java веб-сервер — это **Tomcat**. Названный, кстати, в честь кота Тома из мультфильма “Том и Джери”.

Как же Tomcat взаимодействует с сервлетами? На самом деле этот процесс стандартизирован и называется **жизненным циклом сервлета**. В нем сервлет — это загружаемый объект, а веб-сервер — это **контейнер** сервлетов.

Если сервлет еще не загружен, то:

1. Класс сервлета загружается контейнером.
2. Контейнер создает экземпляр класса (объект) сервлета.
3. Контейнер вызывает метод **init()** у объекта сервлета. Метод вызывается только один раз.

Стандартный цикл работы — **обслуживание клиентского запроса**:

- Каждый запрос обрабатывается в отдельном потоке.
- Контейнер вызывает метод **service()** у сервлета и передает туда объекты ServletRequest и ServletResponse.
- Для завершения работы сервлета вызывается метод **destroy()** у объекта сервлета. Вызывается он всего один раз.

Причин завершения работы сервлета может быть очень много:

- Программист перезапускает веб-сервер, нужно корректно завершить работу всех сервлетов.
- Программист загружает новую версию сервлета, старую нужно правильно выгрузить.
- И так далее.

Запомни главное: веб-сервер и его сервлеты должны без сбоев и перезагрузки работать месяцами, обслуживая тысячи запросов в минуту. Поэтому код и загрузки, и работы, и выгрузки сервлета всегда нужно писать очень качественно.

1.2 Класс HttpServlet

Класс Servlet существует для стандартизации работы сервлета и контейнера. Непосредственно с этим классом программисты не работают. Ну или работают очень редко. Чаще всего используется класс **HttpServlet**, унаследованный от Servlet’a.

У этого класса есть несколько методов, которые нам пригодятся. Ты будешь часто их использовать:

	Метод	Описание
1	<code>init()</code>	Вызывается один раз при загрузке сервлета
2	<code>destroy()</code>	Вызывается один раз при выгрузке сервлета
3	<code>service(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового запроса к сервлету
4	<code>doGet(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового GET-запроса к сервлету
5	<code>doPost(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового POST-запроса к сервлету
6	<code>doHead(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового HEAD-запроса к сервлету
7	<code>doDelete(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового DELETE-запроса к сервлету
8	<code>doPut(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового PUT-запроса к сервлету

Методы `init()` и `destroy()` унаследованы от класса `Servlet`. Поэтому если ты решишь переопределить их в своем сервлете, тебе нужно будет так же вызвать их реализацию из базового класса. Для этого используется команда `super.имяМетода()`.

Пример сервлета:

```
1 public class FirstHttpServlet extends HttpServlet {
2
3     @Override
4     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
5
6         // Получаем параметр “secret” из запроса
7         String secret = request.getParameter("secret");
8
9         // Кладем параметр “secret” в Http-сессию
10        HttpSession session = request.getSession(true);
11        session.setAttribute("secret", secret);
12
13        // Печатаем HTML в качестве ответа для браузера
14        response.setContentType("text/html;charset=UTF-8");
15        PrintWriter out = response.getWriter();
16        try {
17            out.println("<html>");
18            out.println("<head>");
19            out.println("<title>Заголовок</title>");
20            out.println("</head>");
21            out.println("<body>");
22            out.println("<h1>Пример сервлета "+ secret + "</h1>");
23            out.println("</body>");
24            out.println("</html>");
25        } finally {
26            out.close();
27        }
28    }
29 }
```

1.3 Метод `service(HttpServletRequest, HttpServletResponse)`

Если посмотреть на обработку клиентского запроса с точки зрения сервлета, то дела обстоят примерно так.

Для каждого клиентского запроса контейнер (веб-сервер) создает объекты `HttpServletRequest` и `HttpServletResponse`, а затем вызывает метод `service(HttpServletRequest request, HttpServletResponse response)` у соответствующего сервлета. В него передаются эти объекты, чтобы метод мог взять нужные данные из `request`'а и положить результат работы в `response`.

У метода `service()` есть реализация по умолчанию. Если ее не переопределить, то выполняться будет именно она. Вот что он делает.

Метод `service()` определяет из `request`'а тип HTTP-метода (GET, POST, ...) и вызывает метод соответствующий запросу.

	Метод	Описание
1	<code>service(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового запроса к сервлету
2	<code>doGet(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового GET-запроса к сервлету
3	<code>doPost(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового POST-запроса к сервлету
4	<code>doHead(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового HEAD-запроса к сервлету
5	<code>doDelete(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового DELETE-запроса к сервлету
6	<code>doPut(HttpServletRequest, HttpServletResponse)</code>	Вызывается для каждого нового PUT-запроса к сервлету

В своем классе ты можешь или переопределить один метод `service()`, или не трогать его, но тогда переопределять методы `doGet()`, `doPost()`, ... по необходимости.



Комментарии (4)

популярныеновыестарые

JavaCoder

Введите текст комментария

Pixta

Уровень 90

EXPERT

12 августа 2022, 04:39

...

По факту названия Tomcat в честь персонажа мультсериала не нашел никакой информации.

Однако, исходя [официального вики tomcat](#)

Как Tomcat получил свое название?
*Джеймс Дункан Дэвидсон придумал **"Tomcat"**, поскольку животное представляло собой нечто, что могло позаботиться о себе и постоять за себя. Вот так он придумал это имя.*

В других источниках говорится, что было оказано влияние трудов выпуска O'Reilly, имеющие на своих обложках животных. Он был этим так воодушевлен, что хотел тоже выпустить похожее.

[wikipedia](#) трактует, что термин **"Tomcat"** происходит от литературного персонажа XVIII века, кота по имени Том.

Поискав в интернете, нашел полное название **The Life and Adventures of a Cat (1760)**

Ответить

👍 +6 🗨

Никита

Уровень 89

EXPERT

 18 августа 2022, 19:23

⋮

Интересно сделать такой же ресерч на тему почему Сервлет называется Сервлет

Ответить

👍 +2 🗨

Pixta

Уровень 90

EXPERT

 18 августа 2022, 20:20

⋮

Так и появляется осмысление слов. Потом еще апплеты, портлеты))

Ответить

👍 +2 🗨

Anonymous #2436575

Android Developer в AllPets

 6 сентября 2022, 11:47

⋮

Вообще tomcat называют толстых, мощных котов, чаще всего уличных, по которому сразу видно что в драках они явно на районе главные. Потому что общее cat нейтрально по полу и не конкретно. По крайней мере так мне объяснял муж-канадец)
P.S а домашних ухоженных кошек и котят(реже) называют kittycat)

Ответить

👍 +6 🗨

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

