

Настройка сервлета

JSP & Servlets
12 уровень, 4 лекция

ОТКРЫТА

Метод init()

И еще пара полезных мелочей. Конечно, я про инициализацию сервлета. Как ты уже знаешь, после того, как веб-сервер создал объект сервлета и поместил его в контейнер, он вызывает у сервлета метод `init()`. Ты можешь переопределить этот метод и инициализировать в нем все, что тебе нужно.

А почему бы не использовать конструктор?

Потому что процесс создания сервлета происходит примерно так:

- Создаем объект, унаследованный от `HttpServlet`
- Создаем объект `ServletContext`, добавляем его переменную сервлета
- Создаем объект `ServletConfig`, добавляем его переменную сервлета
- ...
- Веб-сервер присоединяет сервлет к контейнеру
- Вызываем метод `init()`

В конструкторе твоего сервлета еще очень много его внутренних переменных не инициализировано. Контейнер ничего не знает о твоём сервлете, твой сервлет ничего не знает о его контексте. Думаю, тут и так все понятно.

Давай напишем сервлет, который при инициализации находит `properties`-файл с настройками:

```
1 public class PropertiesServlet extends HttpServlet {
2     public init() {
3         try (InputStream input = new FileInputStream("c:/path/to/config.properties")) {
4
5             Properties prop = new Properties();
6             prop.load(input);
7
8             String databaseURL = prop.getProperty("db.url");
9             String databaseUser = prop.getProperty("db.user ");
10            String databasePassword = prop.getProperty("db.password");
11        }
12    }
13 }
```

Тут мы создаем объект `Properties` и загружаем в него данные из файла `config.properties`. Ну и в дальнейшем можно доставать из объекта `Properties` различные параметры, такие как данные для доступа к базе данных, например.

Как правильно загрузить properties-файл

Кстати, а что делать, если твой сервлет выполняется не на твоём компьютере?

Скажем, написали его тут, а он выполняется на сервере где-то в другой части света. Или на нескольких серверах. Как правильно загрузить `properties`-файл в этом случае?

Хороший вопрос. Обычно в процессе работы **сервлет знает только относительный путь** своих `properties`-файлов, но не знает их абсолютного пути, так как `war`-файлы с сервлетами могут храниться где угодно.

Значит, нам нужно узнать путь, где храниться наш сервлет (сервлет-то уже проинициализирован) и добавить к нему относительный путь :)

Выглядит это примерно так:

```
1 String path = absoluteServletParh + "относительный путь";
```

И, как всегда, у такой элементарной задачи часто есть свое маленькое “но”. Твой сервлет и его properties-файл хранятся внутри архива :) Не обязательно, конечно, но и такое бывает. properties-файл частенько хранится внутри jar- или war-файлов.

То есть физического пути на диске у твоего файла может не быть. Но так как контейнер сумел загрузить твой сервлет, он скорее всего сможет загрузить и твой properties-файл.

Для этого тебе нужно получить объект загрузчика класса (`ClassLoader`) и попросить его загрузить твой файл. Вот как это будет выглядеть:

```
1 ClassLoader loader = Thread.currentThread().getContextClassLoader();
2 InputStream stream = loader.getResourceAsStream("/config.properties");
3
4 Properties prop = new Properties();
5 prop.load(stream);
```

Метод getConfig()

Кстати, не все параметры можно передавать в сервлет в properties-файлах. Например, твой сервлет взаимодействует с другими сервлетами в распределенном веб-приложении.

Тогда нужно сделать так, чтобы контейнер передал твоему сервлету всю необходимую информацию, когда он будет вызвать его метод `init()`. Более того, он так и делает.

У твоего сервлета (ты ведь помнишь, что он унаследован от класса `HttpServlet`) есть метод `getServletConfig()`. Который возвращает объект `ServletConfig`, созданный и инициализированный контейнером. У этого объекта есть такие методы:

<code>getInitParameterNames()</code>	Возвращает список имен параметров сервлета
<code>getInitParameter(String name)</code>	Возвращает параметр сервлета по его имени
<code>getServletName()</code>	Возвращает собственное имя сервлета
<code>getServletContext()</code>	Возвращает объект <code>ServletContext</code>

Давай напишем сервлет, который возвращает список своих параметров из `ServletConfig` ’а. Класть их туда будет через файл web.xml:

```
<web-app>

    <servlet>
        <servlet-name>Print-Servlet</servlet-name>
        <servlet-class>PrintServlet</servlet-class>
        <init-param>
            <param-name>jdbc-driver</param-name>
            <param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>
        </init-param>
    </servlet>

    <servlet-mapping>
```

```
<servlet-name>Print-Servlet</servlet-name>
<url-pattern>/print</url-pattern>
</servlet-mapping>
```

```
</web-app>
```

Получить свои параметры сервлет может с помощью кода:

```
1 public class PrintServlet extends HttpServlet {
2     public void init() {
3         ServletConfig config = this.getServletConfig();
4         Enumeration<String> initParameterNames = config.getInitParameterNames();
5
6         while (initParameterNames.hasMoreElements()){
7             String key = initParameterNames.nextElement();
8             System.out.println("%s: %s\n", key, config.getInitParameter(key));
9         }
10    }
11 }
```

< Предыдущая лекция

Следующая лекция >

-

+6

+

Комментарии (2)

популярные новые старые

JavaCoder

Введите текст комментария



Никита Уровень 89 EXPERT

19 августа 2022, 17:54



Я где-то пропустил для чего вообще пропертис файл, как он выглядит и тд?

Ответить

-

+3

+

IgorVer System Engineer в Eskaro

6 июля 2022, 13:25



Последний пример не работает. У меня зараюотало только так:
Enumeration<String> initParameterNames = servletConfig.getInitParameterNames();
while (initParameterNames.hasMoreElements()){
String key = initParameterNames.nextElement();
System.out.printf("%s: %s\n", key, servletConfig.getInitParameter(key));
}

Ответить

-

+7

+

👍

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

