

Теги:

**<x:forEach>** – XML-версия тега **<c:choose>**;  
**<x:choose>** – XML-версия тега **<c:forEach>**;  
**<x:if>** – XML-версия тега **<c:if>**;  
**<x:otherwise>** – XML-версия тега **<c:otherwise>**;  
**<x:out>** – XML-версия тега **<c:out>**;  
**<x:param>** – XML-версия тега **<c:param>**, определяющая параметры для другого тега **<x:transform>**;  
**<x:parse>** – разбор XML-документа;  
**<x:set>** – XML-версия тега **<c:set>**;  
**<x:transform>** – трансформация XML-документа;  
**<x:when>** – XML-версия тега **<c:when>**;  
**<x:choose>** – XML-версия тега **<c:choose>**;  
**<x:forEach>** – XML-версия тега **<c:forEach>**.

### Включение ресурсов

В реальных проектах JSP-страницы часто состоят из статических элементов. Для этого используется директива **include**, а файл, содержащий необходимый статичный элемент, сохраняется с расширением **.jspx**, что означает «фрагмент JSP». При необходимости включения содержимого в JSP-страницу каждый раз, когда та получает запрос, используется стандартное действие **jsp:include**. В этом случае включаемые сегменты имеют доступ к объектам **request**, **session** и **application** исходной страницы и ко всем атрибутам, которые имеют эти объекты. Если использовать директиву **include**, то изменения включаемого сегмента отразятся только после изменения исходной страницы (контейнер JSP перекомпилирует исходную страницу). Для включения содержимого в JSP-документ также используется стандартное действие **jsp:include**. При этом не обязательно, чтобы включаемый JSP-фрагмент был правильным XML-документом. Главное, чтобы он возвращал текст в виде правильного XML и не нарушал структуру исходного JSP-документа.

```

<!--пример #15 : включение в код статического содержимого : incl_title.jsx -->
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
          xmlns:c="http://java.sun.com/jsp/jstl/core"
          version="2.0">
  <jsp:directive.page contentType=
    "text/html; charset=UTF-8" />
  <html><head>
    <jsp:directive.include file="\WEB-INF\jspx\title.jsx" />
  </head>
  <body>
    <h1>JSP-страница, использующая директиву include</h1>
    <h3>Директива include используется для включения статиче-
      ского содержимого, например заголовка страницы.</h3>
  </body></html>
</jsp:root>

```

*<!-- пример #16 : код включаемого фрагмента : title.jspf -->*

```
<title>Title from title.jspf</title>
```

Ниже приведен пример включения динамического содержимого. Включаемый фрагмент получает данные из объектов **request** и **session**. Для передачи значения параметра можно использовать строку запроса. Запрос может выглядеть следующим образом:

**http://localhost:8082/home/thanks.jsp?lname=username.**

Установка кодировки в фрагменте необходима для того, чтобы устранить неполадки при включении русского текста.

*<!-- пример #17 : использование действия include для динамического включения : thanks.jspx -->*

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
version="2.0">
```

```
<jsp:directive.page contentType="text/html; charset=utf-8"/>
<html><head><title>Действие include</title></head>
<body>
```

Данные, сформированные динамически при включении jsp-фрагмента<br/>

Включаемый фрагмент получает данные из объекта session <br/>

```
<jsp:include page="\WEB-INF\jspf\params.jsp"/>.
```

```
</body></html>
```

```
</jsp:root>
```

*<!-- пример #18 : включаемый фрагмент : params.jsp -->*

```
<jsp:directive.page contentType="text/html; charset=utf-8"/>
ID сессии -
```

```
<jsp:expression>session.getId()</jsp:expression>
```

В результате работы документа в браузер будет выведено:

**Данные, сформированные динамически при включении jsp-фрагмента.**

**Включаемый фрагмент получает данные из объектов request, session ID сессии - 08C51EEC60A97E90C734101F54EA310E.**

Также для включения содержимого можно использовать тег **<c:import>**. Его использование уже было приведено выше.

## Обработка ошибок

При выполнении web-приложений, как и любых других, могут возникать ошибки и исключительные ситуации. Три основных типа исключительных ситуаций:

- код «404 Not Found». Возникает при неправильном наборе адреса или обращении к странице, которой не существует;
- код «500 Internal Server Error». Возникает при вызове сервлета метода **sendError(500)** для объекта **HttpServletResponse**;
- исключения времени исполнения. Исключения, генерируемые web-приложением и не перехватываемые фильтром, сервлетом или JSP.