

Лабораторная работа №8

Библиотека действий JSTL, страницы обработки ошибок

Цель:

Изучить действия библиотеки JSTL, упрощающие разработку веб-приложений, и получить навыки реализации JSP-страниц не содержащих скриптовых элементов. Изучить механизм страниц обработки ошибок и получить навыки его использования.

Задание:

1. Разработать веб-приложение, содержащее JSP-страницу для вывода таблицы с данными.
2. Использовать в JSP-странице EL-выражения и следующие действия библиотеки JSTL: `<c:out>`, `<c:if>` (или `<c:when>`), `<c:forEach>`, `<c:url>`. Использовать в EL-выражениях функции библиотеки JSTL.
3. Реализовать и проверить функционирование двух страниц обработки ошибок, одна из которых должна быть сопоставлена программному исключению (например, `java.sql.SQLException`), а другая – коду статуса (например, 404 или 500).

Теоретические сведения

Страницы обработки ошибок.

По умолчанию в случае возникновения ошибки времени выполнения при обработке запроса сервер приложений генерирует ответ в виде стандартной страницы с системным сообщением об ошибке и стеком вызовов для возникшего исключения (последовательности вложенных вызовов методов, предшествовавших исключению). Веб-приложение может вместо этого выводить собственную страницу с сообщением об ошибке, понятным конечному пользователю, с помощью механизма страниц обработки ошибок (error pages). Страница обработки ошибок может генерироваться веб-компонентом динамически (сервлетом или JSP-страницей), а может быть простой статической HTML-страницей.

Механизм страниц обработки ошибок работает следующим образом:

1) пользователь делает запрос к прикладному компоненту веб-приложения (например, сервлету-контроллеру);

2) при обработке запроса в прикладном веб-компоненте возникает исключение (например, `java.sql.SQLException`, `javax.ejb.EJBException` и т.д.), которое компонент регистрирует в логе сервера приложений и выбрасывает либо стандартное исключение для технологии сервлетов (`javax.servlet.ServletException`), либо производное от него исключение (например, определенное самим веб-приложением для идентификации различных ошибочных ситуаций);

3) выброшенное прикладным компонентом исключение поступает серверу приложений, и он выполняет поиск страницы обработки ошибок, анализируя настройки в установочном дескрипторе;

4) если страница обработки ошибок для возникшего исключения найдена, то контроллер передает оригинальный запрос и ответ ей так, как если бы был вызван метод `RequestDispatcher.forward()`, при этом:

а) вызовы метода `setStatus()` объектом ответа игнорируются;

б) перед перенаправлением сервер приложений устанавливает приведенные в таблице атрибуты запроса.

Имя атрибута	Тип атрибута	Пояснение
<code>javax.servlet.error.status_code</code>	<code>java.lang.Integer</code>	Код статуса ответа
<code>javax.servlet.error.exception_type</code>	<code>java.lang.Class</code>	Тип возникшего исключения
<code>javax.servlet.error.message</code>	<code>java.lang.String</code>	Системное сообщение об ошибке
<code>javax.servlet.error.exception</code>	<code>java.lang.Throwable</code>	Объект возникшего исключения
<code>javax.servlet.error.request_uri</code>	<code>java.lang.String</code>	URL оригинального запроса
<code>javax.servlet.error.servlet_name</code>	<code>java.lang.String</code>	Имя компонента, выбросившего исключение

Эти атрибуты позволяют странице обработки ошибок сгенерировать адекватный ответ с учетом не только оригинального запроса, но и информации об ошибке, возникшей при его обработке;

5) если страница обработки ошибок для возникшего исключения не найдена, то сервер приложений генерирует стандартную страницу с сообщением об ошибке.

Механизм обработки ошибок может быть задействован и тогда, когда приложение (прикладной веб-компонент) вызывает метод `sendError()` объекта ответа. В этом случае выбор нужной страницы обработки ошибок сервер приложений делает, исходя из кода статуса ответа, указанного как параметр метода `sendError()`.

Из сказанного следует, что в установочный дескриптор можно внедрить таблицу соответствия страниц обработки ошибок различным типам исключений и кодам статуса ответа.

При сопоставлении типа возникшего исключения с таблицей соответствия используется принцип наиболее близкого соответствия. Например, если было выброшено прикладное исключение `ru.johnheadlong.vstore.web.IllegalActionException`, расширяющее стандартное `javax.servlet.ServletException`, а в установочном дескрипторе были назначены страницы обработки ошибок как для `ServletException`, так и для `IllegalActionException`, то будет выбрана последняя.

Если для возникшего исключения не было найдено соответствующей страницы обработки ошибок, то выполняется второй просмотр таблицы соответствий для агрегированного объекта исключения, если оно есть. Агрегировать один объект исключения другим можно, передав агрегируемый объект в качестве параметра конструктора нового исключения, а извлечь его можно методом `getRootCause()`. Пример агрегирования исключения:

```
try {
    return dao.getProductList();
} catch (Exception e) {
    e.printStackTrace();
    throw new ServletException(e);
}
```

В библиотеке JSTL определено действие `<c:catch>`, служащее дополнением к механизму страниц обработки ошибок. Оно позволяет автору страницы обрабатывать те ошибки, которые он может контролировать. Например,

```
<c:catch var="exception">
    <!-- Действия, которые можно пропустить в случае возникновения ошибки -->
    ...
</c:catch>
<c:if test="${exception != null}">
    Извините, обработка не была выполнена по причине...
</c:if>
```

Действие `<c:catch>` обеспечивает автору страницы гранулярную обработку ошибок. Не следует заключать внутрь `<c:catch>` наиболее важные действия на странице, тогда возникающие в них исключения будут направляться на страницы обработки ошибок. Второстепенные же действия рекомендуется заключать в действие `<c:catch>`, чтобы они никогда не приводили к вызову страниц обработки ошибок.

Порядок выполнения работы

1. Откройте или создайте проект веб-приложения. В текущей работе рекомендуется продолжить работу с проектом веб-приложения, созданным в ходе ЛР4. Инструкции по созданию нового проекта веб-приложения приведены в указаниях к ЛР3.
2. Так как в спецификации J2EE 1.4 предоставление библиотеки JSTL не входит в число требований к веб-серверам, необходимо подключить данную библиотеку к проекту веб-приложения. Выберите в дереве проекта узел **Libraries** и выполните в контекстном меню команду **Add Library....** В диалоге добавления библиотеки (рис. 1) выберите строку JSTL 1.1 и нажмите кнопку **Add Library**.

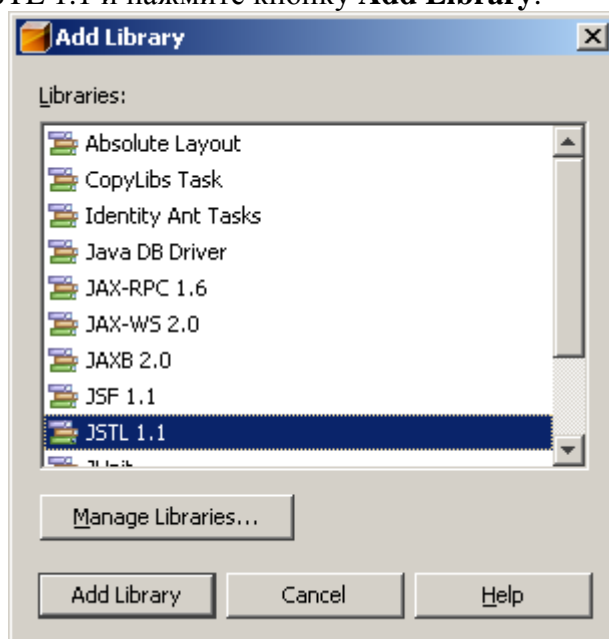


Рис. 1. Добавление библиотеки JSTL

3. Скопируйте страницу, выводящую таблицу с данными при помощи скриптовых элементов, и замените скриптовые элементы на действия библиотеки JSTL и EL-выражения. Не забудьте про директиву `taglib`, необходимую для подключения библиотек действий, определяемых программистом. При этом скорее всего потребуется получать данные из таблицы в сервлет-контроллере и передавать их в JSP-страницу через атрибут запроса. Для добавления действий библиотеки JSTL в текст страницы могут пригодиться элементы палитры, объединенные в группу **JSP**.
4. Запретите использование скриптовых элементов для страницы, созданной в п.2. Для этого откройте в редакторе установочный дескриптор (файл `web.xml`), выберите вкладку **Pages** и нажмите кнопку **Add JSP Property Group....** В диалоге добавления группы JSP-страниц (рис. 2) задайте имя группы и выберите входящие в нее JSP-страницы, нажмите **OK**. После этого в разделе **JSP Property Groups** появляется форма для редактирования созданной группы (рис. 3). Установите флажок **Disable Scripting** и сохраните установочный дескриптор.
5. Запустите приложение, убедитесь в правильности его функционирования.
6. Создайте в проекте JSP-страницы для обработки ошибок. Выводите в них информацию об ошибке, используя описанные выше атрибуты запроса.

7. Возможно, для демонстрации обработки прикладных исключений придется внести изменения в код сервлета-контроллера. Обратите внимание, что описанный механизм страниц обработки ошибок применим только к исключениям, выбрасываемым сервлетом, а не JSP-страницей.
8. Для сопоставления созданных страниц и типов ошибок выберите в редакторе установочного дескриптора закладку **Pages**, раскройте таблицу **Error Pages** и нажмите кнопку **Add...**. В диалоге добавления страницы обработки ошибок (рис. 4) выберите нужную страницу и укажите код статуса либо полностью квалифицированное имя класса исключения, нажмите **OK**. Аналогичным образом задайте соответствие для второй страницы обработки ошибок.
9. Запустите приложение и создайте условия, приводящие к появлению страниц с информацией об ошибках. Предварительно рекомендуется изменить настройку браузера Internet Explorer, которая препятствуют выводу информации об ошибке самим веб-приложением. Для этого откройте свойства Internet Explorer и отключите опцию Advanced / Browsing / Show friendly HTTP error messages.

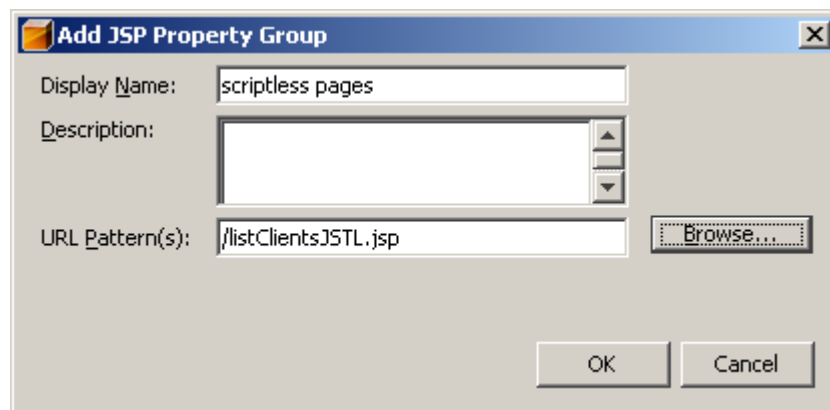


Рис. 2. Добавление группы JSP-страниц

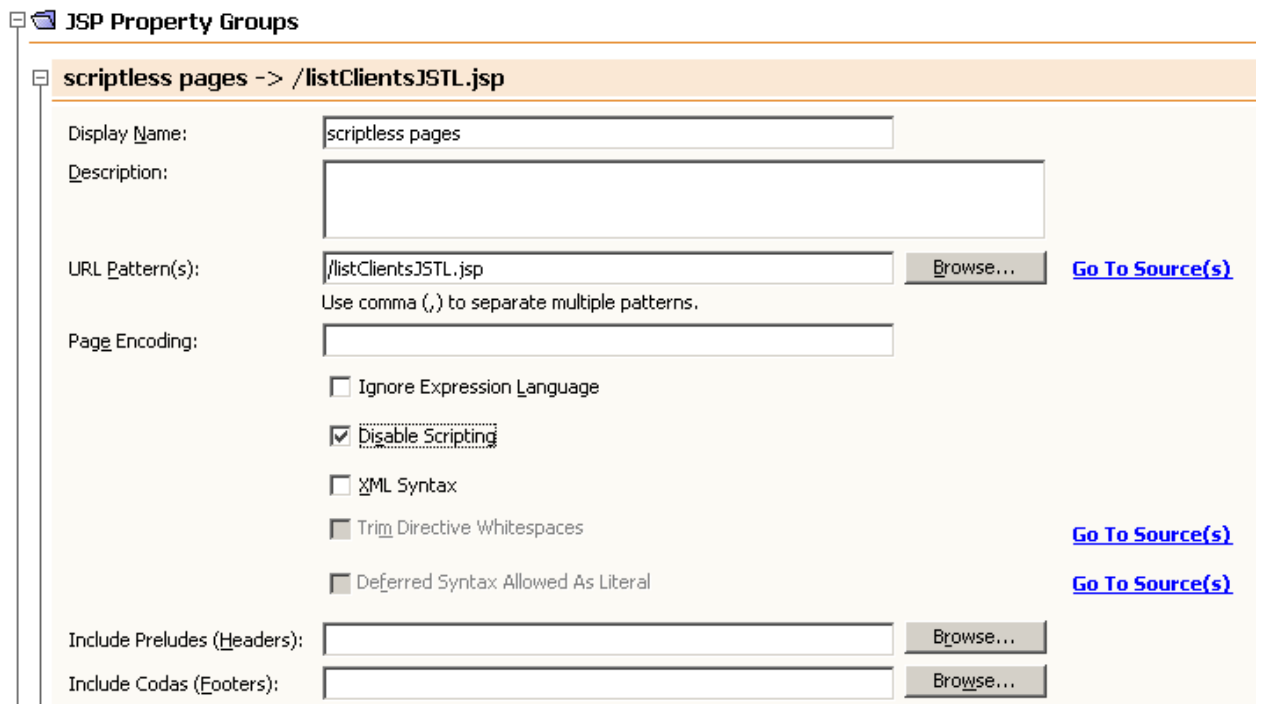


Рис. 3. Редактирование группы JSP-страниц

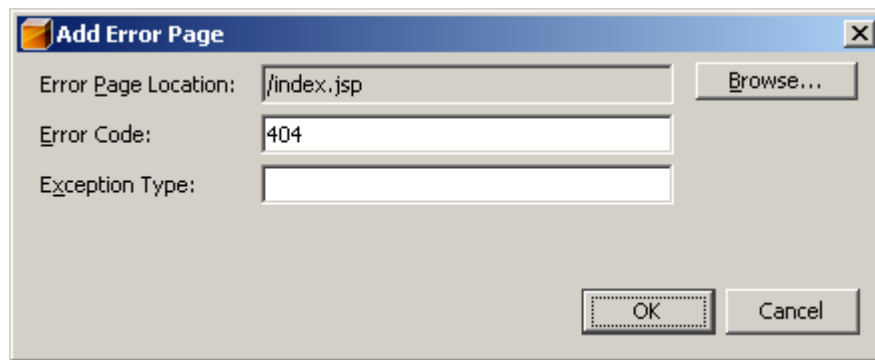


Рис. 4. Добавление страницы обработки ошибок