



Отображение динамической информации на Web-страницах с JavaServer Faces

Содержание

Отображение динамических сведений на Web-страницах с JavaServer Faces . 1

Отображение динамических сведений на Web-страницах с JavaServer Faces	2
Модуль 1: Создание Web-страниц с соединениями обмена данными	4
Урок 1.1: Импорт требуемых ресурсов и задание целевого сервера	5
Урок 1.2: Подключение к базе данных и отображение данных на Web-странице	7
Урок 1.3: Тестирование Web-сайта	13
Урок 1.4: Создание новой записи для отображения и обновления базы данных	15

Урок 1.5: Программирование кнопки Отправить	18
Урок 1.6: Создание и обновление страницы	21
Модуль 1: Итог	25
Модуль 2: Добавление дополнительных функций	26
Урок 2.1: Форматирование таблицы данных	30
Урок 2.2: Применение компонента загрузки файлов	37
Урок 2.3: Применение правил навигации	39
Урок 2.4: Применение функции автоматического создания ключей	41
Модуль 2: Итог	43
Отображение динамических сведений на Web-страницах с JavaServer Faces	44

Отображение динамических сведений на Web-страницах с JavaServer Faces

В этом учебнике рассмотрена процедура разработки динамического Web-сайта, выполняющего роль раздела объявлений газеты. Данный Web-сайт представляет собой полнофункциональное Web-приложение J2EE, применяющее компоненты JavaServer Faces (JSF) и Service Data Objects (SDO). Web-приложение применяет технологию JSF для создания динамических Web-страниц, содержащих ссылки на базу данных с объявлениями. Путем переноса компонентов JSF на страницы можно создать Web-приложение JSF (другое название - приложение CRUD, поскольку оно позволяет создавать, загружать, обновлять и удалять записи базы данных). С помощью этих функций пользователи Web-сайта смогут управлять объявлениями в базе данных.

Цели обучения

Этот учебник разбит на два модуля, каждый из которых обладает своей целью. Модули можно выполнить независимо друг от друга или вместе. Каждый модуль содержит набор упражнений, необходимый для обеспечения правильной работы Web-сайта.

Модуль 1: Создание Web-страниц с соединениями обмена данными

Из данного модуля вы узнаете как настроить соединение с базой данных и использовать информацию из нее на Web-страницах. В этом модуле рассмотрены следующие процедуры:

- Подключение Web-страниц к базе данных.
- Создание страниц, обеспечивающих создание, изменение и удаление записей базы данных.
- Отправки данных из одной страницы в другую

Модуль 2: Добавление дополнительных функций

В данном модуле описаны расширенные способы использования данных из базы данных. В этом модуле рассмотрены следующие процедуры:

- Форматировать записи базы данных на Web-странице
- Добавление компонента, позволяющего загружать файлы в базу данных с Web-страницы.
- Автоматически переходить с одной страницы на другую
- Автоматизировать выполнение некоторых задач администрирования базы данных

Работу с учебником следует начинать с модуля 1: Создание Web-страниц с соединениями с данными.

Необходимое время

Требуемое время: 3 часа 30 минут

Модуль 1: 2 часа

Модуль 2: 1 час 30 минут

Отображение динамических сведений на Web-страницах с JavaServer Faces

В этом учебнике рассмотрена процедура разработки динамического Web-сайта, выполняющего роль раздела объявлений газеты. Данный Web-сайт представляет собой полнофункциональное Web-приложение J2EE, применяющее компоненты JavaServer Faces (JSF) и Service Data Objects (SDO). Web-приложение применяет технологию JSF для создания динамических Web-страниц, содержащих ссылки на базу данных с объявлениями. Путем переноса компонентов JSF на страницы можно создать Web-приложение JSF (другое название - приложение CRUD, поскольку оно позволяет создавать, загружать, обновлять и удалять записи базы данных). С помощью этих функций пользователи Web-сайта смогут управлять объявлениями в базе данных.

Для работы с этим учебником может потребоваться установка дополнительных компонентов. Обратитесь к разделу системных требований и проверьте, все ли дополнительные компоненты установлены.

После выполнения этого учебника пользователи смогут просматривать не только продаваемые предметы, но и добавлять новые предметы, изменять сведения о предметах (например, цену или описание), а также искать предметы определенного типа. Несмотря на простоту примера, показанные в нем принципы и технологии применяются в значительно более сложных Web-сайтах.

Цели обучения

Этот учебник разбит на два модуля, каждый из которых обладает своей целью. Модули можно выполнить независимо друг от друга или вместе. Каждый модуль содержит набор упражнений, необходимый для обеспечения правильной работы Web-сайта.

Модуль 1: Создание Web-страниц с соединениями обмена данными

Из данного модуля вы узнаете как настроить соединение с базой данных и использовать информацию из нее на Web-страницах. В этом модуле рассмотрены следующие процедуры:

- Подключение Web-страниц к базе данных.
- Создание страниц, обеспечивающих создание, изменение и удаление записей базы данных.
- Отправки данных из одной страницы в другую

Модуль 2: Добавление дополнительных функций

В данном модуле описаны расширенные способы использования данных из базы данных. В этом модуле рассмотрены следующие процедуры:

- Форматировать записи базы данных на Web-странице
- Добавление компонента, позволяющего загружать файлы в базу данных с Web-страницы.
- Автоматически переходить с одной страницы на другую
- Автоматизировать выполнение некоторых задач администрирования базы данных

Работу с учебником следует начинать с модуля 1: Создание Web-страниц с соединениями с данными.

Необходимое время

Требуемое время: 3 часа 30 минут

Модуль 1: 2 часа

Модуль 2: 1 час 30 минут

Уровень пользователя

Средний

Аудитория

Разработчики Web-приложений, разработчики пользовательского интерфейса Web-приложений

Системные требования

Для выполнения данного учебника необходимо предварительно установить и настроить рабочий сервер. Данный учебник был протестирован на следующих серверах:

- WebSphere Application Server 6.1

Информация о развертывании SDO на серверах, отличных от WebSphere Application Servers, приведена в разделе справки Развертывание SDO на серверах, отличных от WAS.

Для работы этого учебника могут потребоваться дополнительные функции, которые не были активированы в ходе установки продукта. В случае отсутствия опций пользовательского интерфейса, описанных в учебнике, убедитесь, что включены все требуемые функции (**Окно → Параметры → Возможности**). Для выполнения этого учебника потребуются следующие функции:

- Java Developer
- Web Developer (стандартный)
- XML Developer
- Database Developer
- Enterprise Java Developer
- Данные

Для работы с этим учебником в системе должен быть установлен и настроен сервер приложений. Для того чтобы проверить наличие необходимой рабочей среды сервера, выберите опции **Окно → Параметры**, разверните категорию **Сервер** и нажмите кнопку **Установленные рабочие среды**. С помощью показанной панели вы можете добавлять, удалять и изменять определения установленных рабочих сред сервера. Вы также можете загрузить и установить поддержку нового сервера.

Предварительные требования

Для выполнения данного учебника необходимо обладать знаниями о:

- Основные концепции Web-дизайна, такие как Web-сайты, страницы, браузеры и серверы.
- Создание простой статичной Web-страницы.
- Элементы Web-страниц, такие как таблица, гиперссылка, форма и изображение.
- Термины баз данных, такие как таблица, записи, столбцы и поля

Также нелишним будет понимание следующих принципов:

- Способы применения проекций и панелей в рабочей среде.
- Редактирование кода HTML Web-страницы.

Ожидаемые результаты

По окончании работы над Web-приложением домашняя страница будет выглядеть так, как показано на следующем рисунке. Структура навигации по Web-сайту, добавленная на Web-страницы с помощью шаблона сайта; ссылки на страницы, позволяющие создавать новые объявления и просматривать список объявлений. Действие строки в следующей таблице открывает страницу обновления, позволяющую изменить информацию, связанную с конкретным объявлением.

Home

View all listings »

Post a listing

View all listings

Title	Details	Category
WV Passat wagon 	Description: 2000 good condition Price: \$1,000.00 Phone: 416-512-8665	Automotive
Magic Flute 	Description: Imagine it. Price: \$101.00 Phone: 314-159-265	Entertainment
Video Camera 	Description: Great camera, suitable for amateur or professional use. Price: \$900.00 Phone: 534-333-454	Entertainment
VCR 	Description: Old and worn but still working ok, hence low price. Price: \$40.00 Phone: 322-654-044	Entertainment
Flat Screen TV 	Description: Very clear picture, sound excellent too. Reluctant sale. Price: \$700.00 Phone: 456-456-944	Entertainment
1 2 3 4 5		

Информация, связанная с данной



Просмотрите версию этого учебника в формате PDF.

Модуль 1: Создание Web-страниц с соединениями обмена данными

В этом модуле рассмотрена процедура создания Web-страниц, позволяющих отображать, изменять, удалять и создавать информацию из базы данных. Вы получите представление о технологии JavaServer Faces (JSF) и способах ее применения в сложных операциях.

Цели обучения

Из данного модуля вы узнаете как настроить соединение с базой данных и использовать информацию из нее на Web-страницах. В этом модуле рассмотрены следующие процедуры:

- Подключение Web-страниц к базе данных.
- Создание страниц, обеспечивающих создание, изменение и удаление записей базы данных.

- Отправки данных из одной страницы в другую

Необходимое время

Для выполнения данного модуля потребуется примерно 2 часа. Изучение прочих фасетов динамических Web-сайтов может потребовать больше времени.

Урок 1.1: Импорт требуемых ресурсов и задание целевого сервера

Перед началом работы необходимо импортировать необходимые ресурсы для этого учебника: набор Web-страниц и пример базы данных Derby.

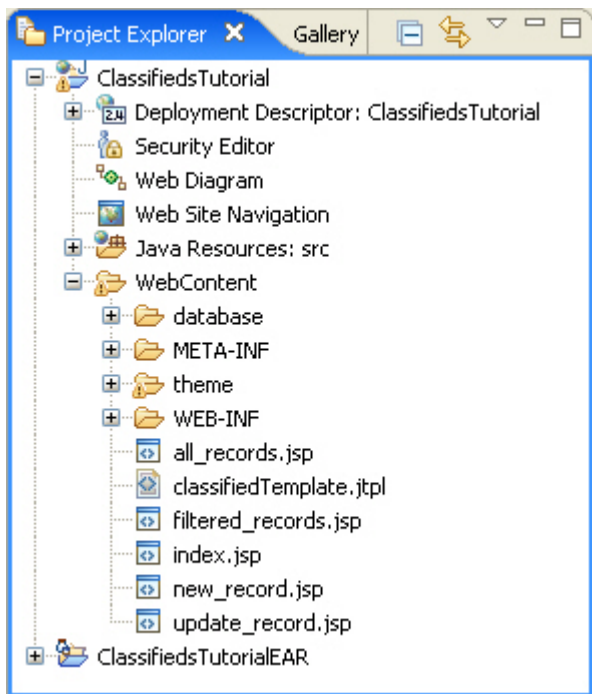
Основная цель этого учебника - обучить вас созданию Web-приложений, предназначенных для работы с данными в базе данных. В нем не затрагиваются вопросы разработки внешнего вида Web-сайта. По этой причине внешний вид Web-сайта был подготовлен заранее.

Для работы с этим учебником учебника вам понадобится доступ к существующему набору файлов JavaServer Faces JSP и пример базы данных Derby. Эти базовые Web-страницы и пример базы данных добавляются в файл ZIP. В данном уроке вы познакомитесь с процессом импорта файла ZIP для применения Web-страниц и баз данных. Кроме того, вы настроите целевой сервер для динамического Web-проекта.

Web-страницы и база данных, необходимые для этого учебника, поставляются в файле ZIP. Для импорта содержимого файла ZIP выполните следующие действия:

Импорт файла примера проекта

1. Импортировать проект. Откройте проекцию Web (выберите **Окно** → **Открыть проекцию** → **Web**).
2. В проекции Web откройте панель Меню проектов и убедитесь, что проект ClassifiedsTutorial выглядит аналогично следующему рисунку:



Задание целевого сервера для динамического Web-проекта

Задав целевой сервер для Web-проекта, вы сможете проверять ресурсы, создаваемые в ходе работы с этим учебником.

Для задания целевого сервера выполните следующие действия:

1. В панели Меню ресурсов проекции Web правой кнопкой мыши щелкните на проекте **ClassifiedsTutorial** и выберите **Свойства**.
2. В списке свойств выберите **Сервер**.
3. В списке Сервер по умолчанию выберите сервер для применения по умолчанию. Нажмите кнопку **Применить**.
4. В списке свойств выберите **Целевые среды выполнения**.
5. В списке Среда выполнения щелкните на среде выполнения, соответствующей выбранному серверу. Нажмите **ОК**.

Если нужный сервер отсутствует в списке целевых сред выполнения, закройте рабочую среду и установите сервер. После установки сервера выполните инструкции по заданию целевого сервера.

Примечание: Если после установки целевых сред выполнения список Сервер по умолчанию пуст, может потребоваться дополнительная настройка сервера. Для настройки сервера выполните следующие действия:

1. Правой кнопкой мыши щелкните на файле **all_records.jsp**, выберите **Выполнить как → Выполнить на сервере**.
2. Выберите **Задать сервер вручную**.
3. Выберите установленный сервер.
4. Настройте сервер, следуя инструкциям мастера. При первом запуске сервера может быть показано сообщение об ошибке. Для исправления ошибки задайте целевой сервер, как это указано выше, перезапустите сервер с помощью панели Серверы, а затем перезагрузите Web-страницу в браузере.

Вернувшись к рассмотренной выше процедуре задания целевого сервера, вы обнаружите, что настроенный сервер выбран в качестве сервера по умолчанию. Если сервер установлен, но не настроен, он не отображается в списке серверов.

Информация о развертывании SDO на серверах, отличных от WebSphere Application Servers, приведена в разделе справки Развертывание SDO на серверах, отличных от WAS.

Контрольные точки урока

Импортирован динамический Web-проект ClassifiedsTutorial и задан целевой сервер.

Вы можете просмотреть файлы Web-проекта учебника. Чтобы открыть файл дважды щелкните на нем на панели Проводник проектов. Для просмотра графического представления связей между страницами дважды щелкните на записи Навигация по Web-сайту в панели Меню проектов.

В этом примере вы, в основном, будете работать со следующими файлами:

all_records.jsp

Домашняя страница Web-сайта. На ней отображаются все объявления из базы данных.

new_record.jsp

Страница, позволяющая создать новое объявление.

update_record.jsp

Страница, позволяющая изменить или удалить объявление из базы данных.

classifiedTemplate.jtpl

Шаблон страниц сайта. Он содержит такие элементы, как таблица и серый баннер "Добро пожаловать в службу объявлений", отображаемый на каждой странице. Кроме того, эта страница содержит две навигационные вкладки под баннером, позволяющие перейти к домашней странице и к странице добавления нового объявления.

Теперь вы можете приступить к упражнению 1.2: Работа с реляционными списками записей и компонентами таблицы данных.

Урок 1.2: Подключение к базе данных и отображение данных на Web-странице

В этом уроке вы узнаете как подключаться к базе данных и отображать текстовые записи на Web-странице. Кроме того, вы научитесь добавлять на Web-страницу список реляционных записей.

На Web-сайте в данном учебнике для доступа к базе данных и отображения информации на странице используются динамические Web-страницы с компонентами JavaServer Faces. В этом учебнике используются реляционные записи и списки реляционных записей для представления данных в базе данных чтобы данные можно было просмотреть на странице в виде таблицы данных или обычной таблицы HTML. Этими компонентами используются объекты доступа Java.

Дополнительные сведения об объектах данных:

Объектами доступа к данным являются объекты Java, представляющие объекты EJB. Обычно они применяются в клиентских программах, работающих с файлами JSP, сервлетами или EJB, взаимодействующими с другими EJB. Объекты доступа позволяют вам не сталкиваться со сложностями управления жизненными циклами объектов EJB. Это означает, что вы можете программировать объекты EJB также легко, как объекты Java, что упрощает клиентские программы объектов EJB и сокращает общее время разработки.

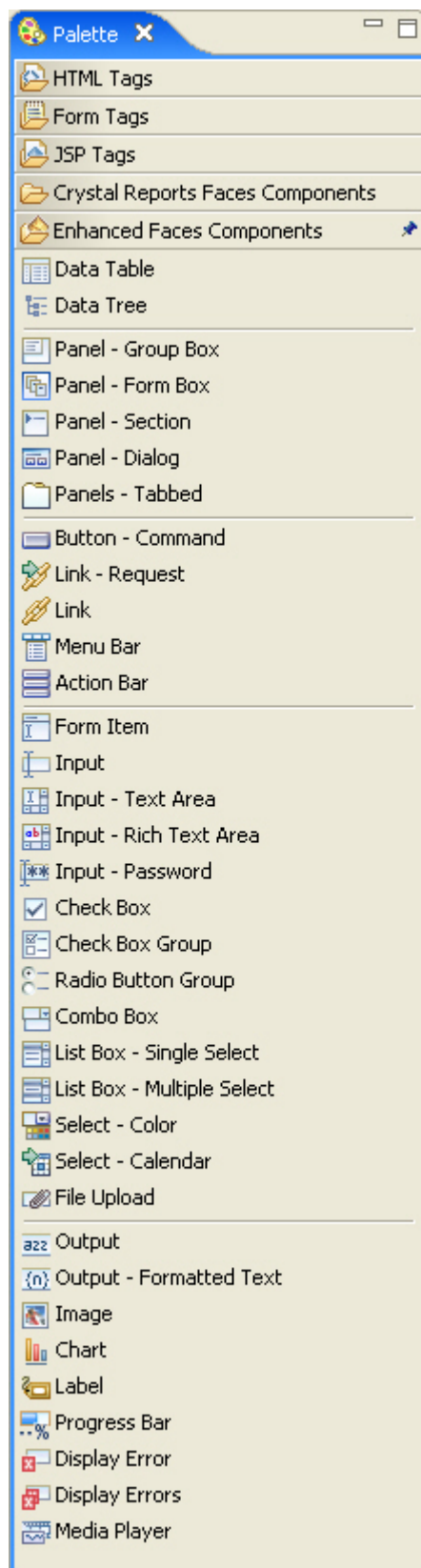
Дополнительная информация о компонентах JavaServer Faces и Faces:

JavaServer Faces - это технология, предназначенная для разработки пользовательских интерфейсов динамических Web-приложений, работающих на сервере. Среда JavaServer Faces управляет состоянием пользовательского интерфейса с помощью серверных запросов и предоставляет простую модель для разработки серверных событий, активируемых клиентом. Технология JavaServer Faces создана на основе среды модель-представление-контроллер (MVC). В контексте JavaServer Faces это означает, что роль контроллера выполняет сервлет, модель представлена объектами EJB, а представление состоит из компонентов JavaServer Faces и кода приложения. Цель этой модели - создание границы между содержимым и представлением.

Такие инструменты, как компоненты Faces, позволяют эффективно применять эту технологию в Web-приложениях. В состав компонентов Faces входит библиотека нестандартных тегов JavaServer Page (JSP), обеспечивающая выражение интерфейса JavaServer Faces на странице. Данный мастер предназначен для создания файлов JSP с поддержкой компонентов Faces. С помощью компонентов Faces Web-приложение можно разрабатывать путем переноса на создаваемые Web-страницы компонентов из лотка Faces панели Палитра.

Например, на страницу можно положить текстовое поле ввода. Затем вы можете перенести кнопку отправки и положить ее рядом с текстовым полем ввода. Наконец, вы можете соединить текстовое поле ввода с источником данных. В результате пользователи Web-приложения смогут вводить данные в источник данных.

Другое преимущество приложений, созданных с помощью компонентов Faces, заключается в том, что пользовательский интерфейс выводится независимо от логики программы. Такой подход позволяет запускать приложения на различных платформах, например, браузерах и портативных устройствах.



Добавление списка реляционных записей

В этом уроке вы создадите список реляционных записей для представления всех объявлений в базе данных. Затем вы подключитесь к базе данных и выберите таблицу, содержащую сведения, необходимые для списка реляционных записей. Наконец вы добавите список реляционных записей на страницу в таблице данных.

- Реляционные записи подключаются только к одной записи из базы данных. В таком случае реляционная запись представляет одно объявление из базы данных. С помощью реляционной записи можно создать одну запись, изменить существующую запись или удалить старую запись.
- Список реляционных записей подключается к нескольким записям из базы данных. В таком случае список реляционных записей представляет от одного до всех объявлений из базы данных.
- В таблицах данных отображаются данные из списка реляционных записей на странице Таблицы данных просто являются местом хранения списков записей, они не форматируют данные в строки и столбцы, как это делает таблица HTML.

Дополнительные сведения о таблицах данных:

Таблица данных является компонентом на странице Faces JSP, содержащим объекты данных, такие как список реляционных записей. Несмотря на то, что таблица с данными имеет строки и столбцы, наподобие таблицы HTML, она не является таблицей в полном смысле этого слова. При необходимости форматирование управляющих элементов ввода и вывода, вам понадобится поле Панель - Группа из лотка Расширенные компоненты Faces на палитре. Этот вопрос описан в разделе Упражнение 2.2: Форматирование таблицы данных.

Так как таблицы данных являются компонентами Faces, а не компонентами HTML, они управляются с помощью панели Свойства, а не панели Проектировщик страниц. С помощью панели Свойства можно настроить таблицу данных различными способами:

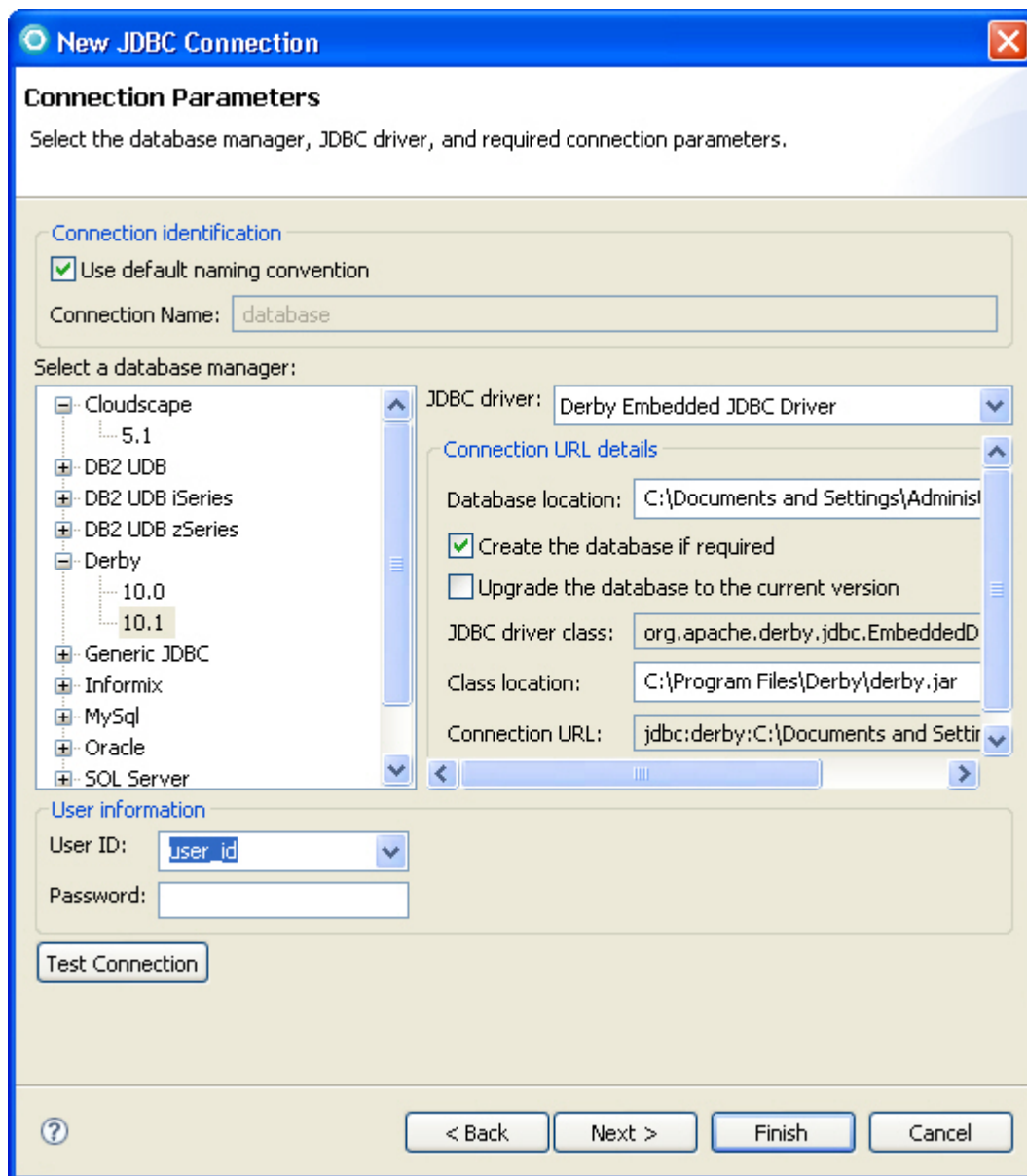
- Добавить, удалить и изменить расположение столбцов
- Форматировать текст и фон
- Добавить заголовок, нижний колонтитул и поля
- Добавить страницы для отображения результатов

Для создания списка реляционных записей:

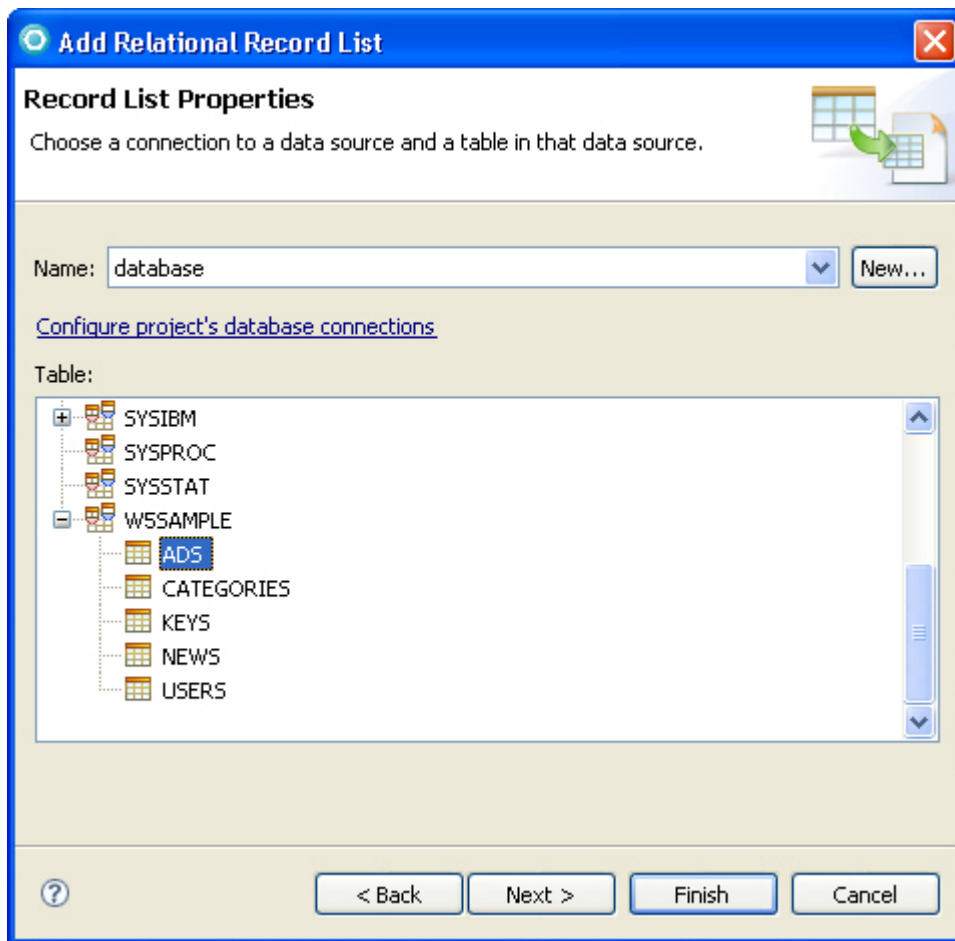
1. В панели Структура проекта проекции Web разверните **ClassifiedsTutorial** → **WebContent**.
2. В папке Содержимое Web-страниц дважды щелкните на файле **all_records.jsp**. В редакторе откроется файл **all_records.jsp**.
3. Удалите текст по умолчанию Содержимое тела по умолчанию.
4. На панели Палитра разверните **Лоток данных**.
5. Перенесите компонент **Список реляционных записей** из палитры в пустую область содержимого. Откроется окно Добавить список реляционных записей.
6. В поле Имя введите **all_recordlist**. Имена списка реляционных записей и самих реляционных записей должны соответствовать стандарту именования имен переменных Java, например, они не могут содержать пробелы.
7. Убедитесь, что выбрана опция **Добавить управляющие элементы данных**. Если опция **Добавить управляющие элементы данных** выбрана, мастер создаст таблицу данных для отображения списка записей на странице. В противном случае мастер создаст только список записей без представления данных на странице. Сейчас же мастер создаст таблицу данных по умолчанию, которую вы сможете настроить в дальнейшем. Окно Добавить список реляционных записей будет выглядеть следующим образом:

Нажмите кнопку **Далее**.

8. На следующей странице мастера в поле **Имя соединения** нажмите кнопку **Создать** для создания нового соединения с базой данных. Откроется окно Создать соединение JDBC. Выберите переключатель **Создать новое соединение** и нажмите кнопку **Далее**.
9. В списке **Выбрать администратора базы данных** разверните запись **Derby** и выберите **10.1**.
10. В поле **Расположение базы данных** нажмите кнопку **Обзор** и выберите `<расположение-рабочей-области>\ClassifiedsTutorial\WebContent\database`, где `<расположение-рабочей-области>` - это каталог текущей рабочей области. Нажмите кнопку **ОК**, чтобы закрыть окно **Выбрать папку**.
11. В поле **Расположение класса** введите расположение одного или нескольких файлов JAR с реализацией класса драйвера JDBC, либо нажмите кнопку **Обзор**, чтобы выбрать нужное расположение в файловой системе. После выбора расположения файлов JAR нажмите кнопку **Открыть**.
12. Для обращения к базе данных может потребоваться указать ИД пользователя. Пароль не требуется. Мастер Создать соединение с базой данных выглядит следующим образом:



13. В мастере Создать соединение JDBC нажмите кнопку **Готово**. Вы вернетесь к мастеру Добавить список реляционных записей.
14. Выберите опцию **Настроить соединения проекта с базой данных**. Будет показано окно свойств соединений JDBC.
15. Выберите опцию **Изменить** рядом с разделом сведений о рабочих соединениях. В качестве типа соединения выберите **Использовать соединение диспетчера драйверов**. Для возврата к окну Добавить список реляционных записей нажмите кнопку **Готово**, а затем **ОК**. Теперь, создав соединение с базой данных Derby, необходимо выбрать таблицу или список записей для представления. Мастер Добавить список реляционных записей покажет таблицы из базы данных.
16. Разверните запись **W5SAMPLE** и выберите **ADS**. Нажмите кнопку **Далее**. На оставшихся страницах мастера можно добавить столбцы для списка записей и настроить дополнительные опции, например задать главный ключ, взаимосвязи с другими таблицами и указать фильтры и критерии сортировки. Последующие упражнения помогут вам научиться работать с этими страницами.



17. На странице Добавить реляционную запись укажите столбцы, которые требуется добавить в список реляционных записей. Нажмите кнопку **Далее**.
18. Для данного учебника вам понадобится отображение лишь заголовка, описания, категории, цены и телефонного номера для каждого объявления. По умолчанию выбраны все столбцы; отмените выбор всех столбцов за исключением следующих:

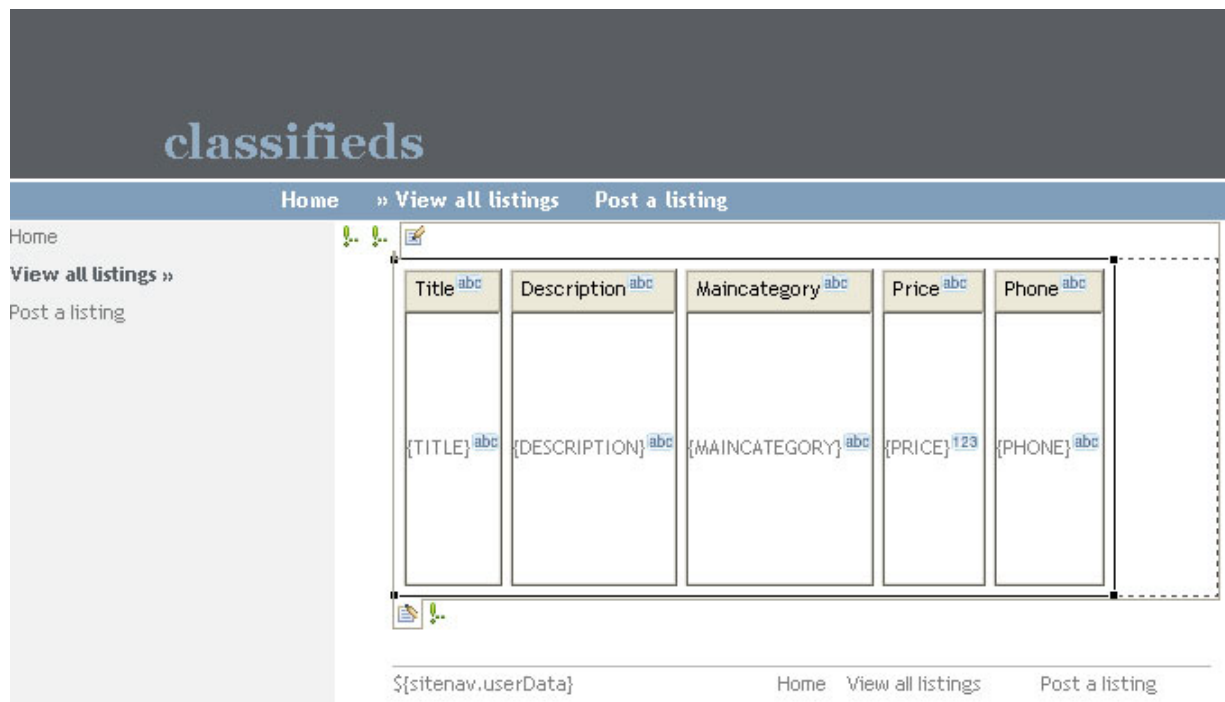
- **ГЛАВНАЯ КАТЕГОРИЯ:** String
- **ЗАГОЛОВОК:** String
- **ОПИСАНИЕ:** String
- **ЦЕНА:** BigDecimal
- **ТЕЛЕФОН:** String

19. Можно дополнительно уточнить организацию отображения таблицы данных на Web-странице. Для изменения порядка столбцов выберите столбец и переместите его в нужное положение с помощью кнопок со стрелками вверх и вниз.

Расположите столбцы в следующем порядке:

- **ЗАГОЛОВОК**
- **ОПИСАНИЕ**
- **ГЛАВНАЯ КАТЕГОРИЯ**
- **ЦЕНА**
- **ТЕЛЕФОН**

Нажмите кнопку **Готово**. Теперь Web-страница выглядит следующим образом:



20. Сохраните страницу.

В следующем уроке вы узнаете как будет выглядеть эта страница после публикации на Web-сервере. Для форматирования таблиц с данными и компонентов JavaServer Faces существует несколько вариантов. Некоторые из них будут описаны в следующем модуле. Можно также просмотреть панель Свойства для различных компонентов JavaServer Faces (например, таблицы с данными и отдельные компоненты ввода).

Контрольные точки урока

Вы закончили урок 1.2. Вы научились подключаться к базе данных и отображать текстовые записи на Web-странице с помощью списка реляционных записей.

Урок 1.3: Тестирование Web-сайта

Когда ваше Web-приложение будет готово к опубликованию, вам потребуется сервер для хостинга, чтобы пользователи смогли обращаться к Web-сайту через Internet, однако для тестирования сайта можно воспользоваться средой тестирования.

В данном уроке вы выполните JSP в среде тестирования для проверки работы и внешнего вида страницы после публикации. Вы также научились запускать и останавливать сервер. В любое время в ходе разработки Web-сайта вы сможете открыть страницу в проектировщике страниц и на вкладке **Предварительный просмотр** посмотреть как она будет выглядеть в браузере. Однако, панель Предварительный просмотр не позволяет проверить динамические аспекты страницы (например, соединение с базой данных) и узнать, как они будут выполняться на сервере.

Совет: Запустив сервер необходимо остановить его перед продолжением работы с сайтом.

Для работы с этим учебником в системе должен быть установлен и настроен сервер приложений. Для того чтобы проверить наличие необходимой рабочей среды сервера, выберите опции **Окно → Параметры**, разверните категорию **Сервер** и нажмите кнопку **Установленные рабочие среды**. С помощью показанной панели вы можете добавлять, удалять и изменять определения установленных рабочих сред сервера. Вы также можете загрузить и установить поддержку нового сервера.

Для тестирования Web-сайта на сервере выполните следующие действия:

1. В панели Меню проектов правой кнопкой мыши щелкните на файле **all_records.jsp**, выберите **Выполнить как** → **Выполнить на сервере**. Откроется окно Выполнить на сервере.
2. Необходимо использовать тот же сервер, что и выбранный целевым в уроке 1.1. Применение другого сервера вызовет ошибки.
 - Для тестирования Web-сайта в ранее настроенной конфигурации сервера выполните следующие действия:
 - a. Выберите опцию **Выбрать существующий сервер**.
 - b. Выберите нужный сервер в списке доступных серверов.
 - Для тестирования Web-сайта в новой конфигурации сервера выполните следующие действия:
 - a. Выберите опцию **Задать новый сервер вручную**.
 - b. Выберите нужный тип сервера в списке.
3. Нажмите кнопку **Готово**.

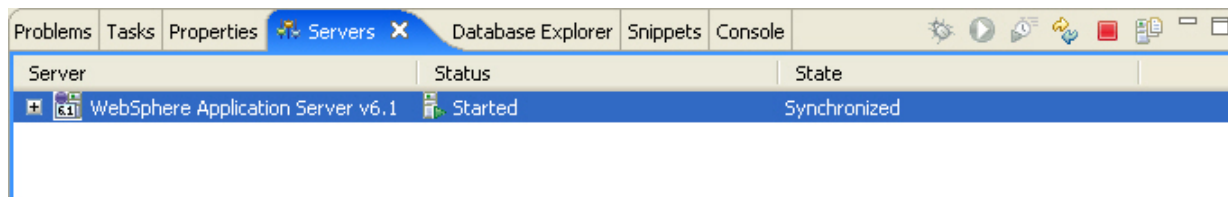
Будет создан новый сервер, запущен и страница откроется во внутреннем Web-браузере рабочей среды. Это может занять пару минут. На панели Консоль отображаются сообщения в ходе запуска сервера.

Теперь вы можете заранее проверить, как будет выглядеть и работать сайт после его опубликования на Web-сервере. Можно щелкать на ссылках, вводить данные в формы и просматривать динамически создаваемое содержимое сайта, которое может не отобразиться на панели Предварительный просмотр. Также при необходимости проверить страницу в различных Web-браузерах, можно скопировать URL из Web-браузера рабочей среды в любой другой браузер.

Перед продолжением выполнения упражнений необходимо остановить сервер. Если оставить сервер работающим, при продолжении выполнения упражнений возникнут ошибки. Эти ошибки возникают из-за того, что сервер подключен к примеру базы данных, что мешает подключиться к базе данных для изменения информации, отображаемой на Web-сайте.

Для остановки сервера:

1. Перейдите на вкладку Серверы. Эта панель обычно расположена внизу страницы по центру. Если панель Серверы не показана, выберите **Окно** → **Показать панель** → **Серверы**.
2. Выберите сервер, который требуется остановить. Убедитесь, что для сервера указано состояние *Запущен*.



3. Нажмите кнопку **Остановить сервер** (красный квадратик) в верхнем правом углу панели Серверы. В поле **Состояние** будет указано значение *Останавливается*. На панели Консоль появятся сообщения, отражающие процесс завершения работы сервера. При изменении состояния на панели Серверы на **Остановлен**, сервер будет остановлен и вы сможете продолжить работать с Web-сайтом.

Совет: вы можете проверить Web-сайт на сервере, но не забудьте после этого остановить сервер.

Контрольные точки урока

Вы закончили урок 1.3. В данном уроке в среде тестирования был выполнен JSP для проверки работы и внешнего вида страницы после публикации. Кроме того, вы научились запускать и останавливать сервер.

Урок 1.4: Создание новой записи для отображения и обновления базы данных

В этом уроке вы создадите страницу, позволяющую пользователям создавать новые объявления и помещать их в базу данных.

Вначале создайте реляционную запись, представляющую новую запись в базе данных, затем создайте визуализацию записи на странице в виде набора полей ввода.

Для создания новой реляционной записи выполните следующие действия:

1. Дважды щелкните на файле **new_record.jsp** на панели проводника проектов чтобы открыть его.
2. Удалите текст по умолчанию Содержимое тела по умолчанию.
3. На панели Палитра щелкните на блоке **Данные** чтобы развернуть его.
4. Переместите компонент **Реляционная запись** из палитры в пустую область содержимого. Появится окно Добавить реляционную запись.

Примечание: Появление предупреждения с сообщением о невозможности установить соединение с базой данных означает, что вы оставили сервер работающим после тестирования Web-сайта. В таком случае нажмите кнопку **Отмена** во всех окнах и остановите сервер, как описано в разделе Урок 1.3.

5. В поле **Имя** введите `create_record`.
6. В пункте **Создать управляющие элементы для** выберите **Создание новой записи**.
7. Убедитесь, что опция **Добавить управляющие элементы ввода-вывода для отображения записи на web-странице** выбрана. Окно Добавить реляционную запись выглядит следующим образом:

Add Relational Record

Relational Record
Enter a reference name for record.

Name:

Create a name to refer to this record within the page.

☐ Retrieve an existing record or record list from scope

Scope :

Key :

☐ Reuse metadata definition from an existing record or record list

Input file:

You can automatically add data controls to your page to work with this record (configure details on last page of this wizard).

☒ Add input/output controls to display the record on the web page:

Create controls for:

☐ Displaying an existing record (read-only)

☐ Updating an existing record

☒ Creating a new record

8. Нажмите кнопку **Далее**.
9. В поле Таблица разверните запись **W5SAMPLE** и выберите ADS. Нажмите кнопку **Далее**.
10. Снова нажмите кнопку **Готово**. Откроется страница Настроить управляющие элементы данных.

Совет: На странице Настроить управляющие элементы данных можно настроить визуализацию реляционных записей. Например, можно изменить столбцы, метки полей и кнопку отправки формы ввода. Выполнив эти действия вы получите полнофункциональную страницу ввода.

11. В области **Отображаемые поля** снимите выделение с переключателями рядом с именем поля, за исключением тех, которые должны присутствовать в форме ввода:
 - ОПИСАНИЕ
 - ИД
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ТЕЛЕФОН
 - ЦЕНА
 - НАЗВАНИЕ
12. Измените порядок размещения полей с помощью кнопок Вверх и Вниз:
 - ИД

- НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН
- Измените метки. Например, можно сократить метку "Главная категория:" до "Категория:". Для того чтобы изменить метки, созданные для полей ввода, щелкните на столбце **Метка**. Значок указателя мыши превратится в курсор и вы сможете ввести текст.
 - Выберите **ГЛАВНАЯ КАТЕГОРИЯ**, затем выберите **Опции**. Откроется окно Опции.
 - Убедитесь, что опция **Кнопка отправки** выбрана.
 - Введите Добавить новый список в поле **Метка** Нажмите кнопку **ОК**. Окно Добавить реляционную запись выглядит следующим образом:

Add Relational Record

Configure Data Controls

Specify the columns to display and how to display them

Fields to display:

Field Name	Label	Control Type
☒ ID (java.lang.Integer)	Id:	Input Field
☒ TITLE (java.lang.String)	Title:	Input Field
☒ DESCRIPTION (java.lang.String)	Description:	Input Field
☒ MAINCATEGORY (java.lang.String)	Category:	Input Field
☒ PRICE (java.math.BigDecimal)	Price:	Input Field
☒ PHONE (java.lang.String)	Phone:	Input Field
☐ TYPECODE (java.lang.Short)	Typecode:	Input Field
☐ DATEOPEN (java.util.Date)	Dateopen:	Input Field
☐ CATEGORY_ID (java.lang.Integer)	Category_id:	Input Field
☐ TYPE (java.lang.String)	Type:	Input Field
☐ EMAIL (java.lang.String)	Email:	Input Field
☐ PHOTO (byte[])	Photo:	File Upload
☐ IMAGETYPE (java.lang.String)	Imagetype:	Input Field
☐ LOCATION (java.lang.String)	Location:	Input Field

- Нажмите кнопку **Готово** для создания формы ввода. Она должна выглядеть следующим образом:

The screenshot shows a web application interface for posting a listing. The main header is 'classifieds'. The navigation bar includes 'Home', 'View all listings', and '» Post a listing'. The sidebar on the left has 'Home', 'View all listings', and 'Post a listing »'. The main content area contains a form with the following fields:

Id:	{ID}	123
Title:	{TITLE}	abc
Description:	{DESCRIPTION}	abc
Category:	{MAINCATEGORY}	abc
Price:	{PRICE}	123
Phone:	{PHONE}	abc

Below the form is a button labeled 'Post New Listing' and a section for '[Error Messages]'. At the bottom of the page, there is a footer with the text '\${sitenav.userData}' and navigation links 'Home', 'View all listings', and 'Post a listing'.

Примечание: В форме есть поле Сообщения об ошибках. Это не значит что в проекте присутствуют ошибки, это поле помечает место, где будут показаны сведения об ошибках при отправке формы пользователем.

18. Сохраните страницу.

При отправке формы вода, страница автоматически добавит новую запись в базу данных. В следующем уроке мы запрограммируем кнопку **Добавить новый список** для возврата к странице `all_records.jsp`, чтобы можно было немедленно просмотреть новую запись в базе данных.

Контрольные точки урока

Вы закончили урок 1.4. В этом уроке научились создавать страницу, позволяющую пользователям создавать новые объявления и помещать их в базу данных.

Урок 1.5: Программирование кнопки Отправить

При отправке формы вода, страница автоматически добавит новую запись в базу данных. В этом уроке вы запрограммируете кнопку **Добавить новый список**. Эта кнопка позволяет вернуться к странице `all_records.jsp` и сразу же просмотреть новую запись в базе данных.

Для настройки кнопки Отправить выполните следующие действия:

1. В файле `new_record.jsp` правой кнопкой щелкните на кнопке **Добавить новый список**, расположенной в форме на Web-странице, и выберите **Свойства**.
2. Нажмите кнопку **Добавить правило**. Откроется окно Добавить правило навигации.
3. В выпадающем списке Страница выберите **all_records.jsp** и нажмите кнопку **OK**.
4. Сохраните страницу. Рекомендуется запустить на тестовом сервере файл `new_record.jsp`, чтобы добавить запись в базу данных и проверить возврат к странице `all_records.jsp`.

Избегайте повторяющихся ключей

Важное замечание:

Поскольку столбец ИД представляет собой первичный ключ таблицы ADS, в таблицу нельзя добавить записи с существующим значением ИД. В модуле 2 рассмотрена функция автоматического создания ключей, с помощью которой для новых записей можно создавать незанятые ключи.

Однако в данном модуле для добавления новой записи следует вручную указать незанятое значение столбца ИД. Записи, поставляемые вместе с базой данных, содержат ИД с номерами от 1 до 24, так что вы можете ввести любое число больше 24 в качестве главного ключа. В случае ввода нескольких записей убедитесь в отсутствии совпадающих ключей.

Связывание ввода с реляционной записью (необязательно)

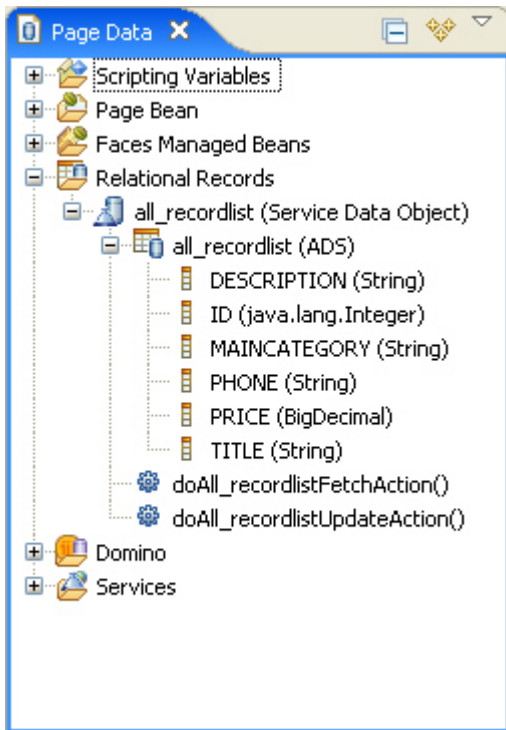
Форма ввода представляет собой набор полей ввода JavaServer Faces, связанных с созданной реляционной записью. Вспомните, что реляционная запись была создана для представления новой записи в таблице ADS базы данных. Связывание - это метод, предусматривающий создание связи между компонентом ввода JavaServer Faces и столбцом реляционной записи.

После создания реляционной записи мастер автоматически связывает все ее столбцы с полями ввода на странице. При необходимости отдельные столбцы можно вручную связать с другими полями ввода. Для связывания столбца реляционной записи с полем ввода с помощью указателя мыши перенесите столбец из панели Данные страницы и положите его на нужное поле. Рекомендуется ознакомиться с этим процессом на практическом примере путем удаления и повторного создания поля ввода Описание.

Дополнительная информация о панели Данные страницы:

Как правило, панель Данные страницы расположена в нижнем левом угле рабочей среды. Если она не показана, выберите **Окно → Показать панель → Данные страницы**. Как правило, панель Данные страницы применяется только при создании динамических Web-страниц.

На панели Данные страницы соединения с источниками данных представлены в виде объектов данных, таких как реляционные записи или списки реляционных записей. Эти объекты данных не представляют ни сам источник данных, ни какой-либо компонент на странице. Эти объекты данных представляют соединение между проектом и источником данных. После создания соединения объект данных можно перенести с панели Данные страницы на Web-страницу для завершения соединения и отображения информации из источника данных на Web-странице. На следующем рисунке показана панель Данные страницы со списком реляционных записей, созданных в этом учебнике:



На этом рисунке показан список реляционных записей `all_recordlist (ADS)` со всеми столбцами. После того, как объект данных определен и показан на панели Данные страницы, эти столбцы можно перенести на компоненты ввода Web-страницы произвольное число раз без повторного подключения к базе данных.

Например, если на Web-странице требуется показать значение столбца ОПИСАНИЕ, сначала следует перенести на страницу текстовое поле вывода, а затем перенести на это поле столбец ОПИСАНИЕ из панели Данные страницы. Такой процесс называется связыванием столбца с полем вывода.

Или, если вы хотели позволить пользователю ввести новое значение в столбце ОПИСАНИЕ в базе данных, вы можете перенести текстовое поле ввода на страницу. Затем перенесите столбец ОПИСАНИЕ на это текстовое поле ввода.

Следующая процедура приведена для дополнительного описания концепции поля ввода и процесса связывания; однако выполнять ее необязательно. Вы можете приступить к следующему уроку.

1. В файле `new_record.jsp` щелкните на поле ввода **Описание**. Нажмите клавишу **Удалить**.
2. В панели Палитра разверните лоток **Расширенные компоненты Faces**.
3. С помощью указателя мыши перенесите компонент **Ввод** с панели Палитра на ячейку, из которой ранее было удалено поле ввода **Описание**. Теперь в этой ячейке расположено поле ввода, однако в нем не показан текст (например, {ИД} или {ЗАГОЛОВОК}), поскольку оно не связано ни с одним из столбцов.
4. В панели Данные страницы выберите **Реляционные записи** → **create_record (Service Data Object)** → **create_record (ADS)** и перенесите столбец ОПИСАНИЕ на созданный компонент **Ввод**. Внутри компонента **Ввод** будет показан текст, указывающий на связывание со столбцом ОПИСАНИЕ. См. следующий рисунок:

Id:	{ID} 123
Title:	{TITLE} abc
Description:	{DESCRIPTION} abc
Category:	{MAINCATEGORY} abc
Price:	{PRICE} 123
Phone:	{PHONE} abc
{Error Messages}	
Post New Listing	

5. Сохраните файл и запустите страницу на тестовом сервере.

Контрольные точки урока

Вы закончили урок 1.5. В этом уроке вы научились настраивать кнопку команды и связывать данные с полем ввода путем переноса виджета JSF с палитры на Web-страницу.

Урок 1.6: Создание и обновление страницы

На этом этапе уже созданы страницы, позволяющие просматривать и создавать списки объявлений. В этом упражнении создается страница, с помощью которой пользователи могут обновлять и удалять списки. Страница обновления выглядит аналогично странице создания; различие заключается в том, что на странице обновления в полях ввода отображаются данные из существующей записи, выбранной для изменения.

Сначала необходимо создать реляционную запись, представляющую существующую запись из базы данных. Затем для этой реляционной записи создается форма обновления JavaServer Faces и после нескольких дополнительных изменений страница будет готова к тестированию.

Создание реляционной записи обновления

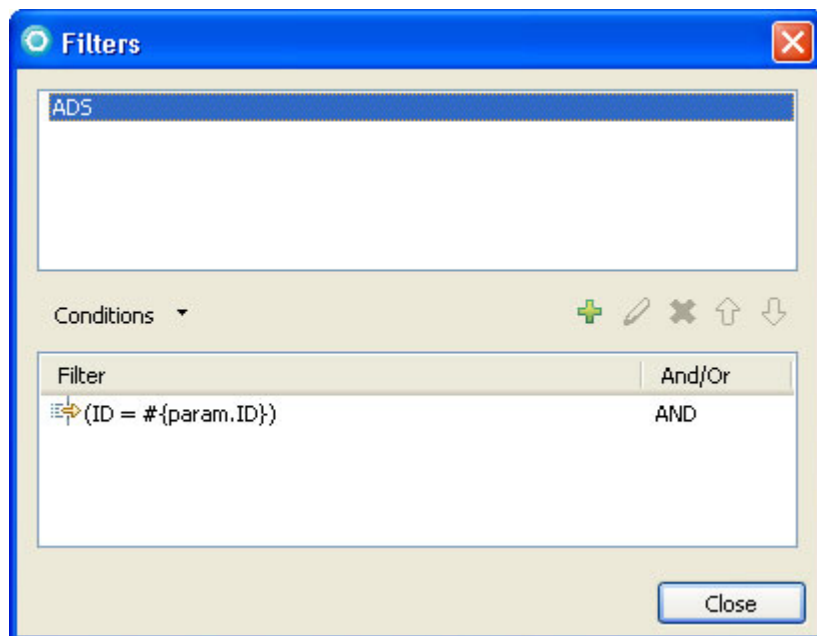
1. Дважды щелкните на файле **update_record.jsp** в панели Меню проектов.
2. Удалите текст по умолчанию Содержимое тела по умолчанию.
3. На панели Палитра щелкните на блоке **Данные** чтобы развернуть его.
4. Переместите компонент **Реляционная запись** из палитры в пустую область содержимого. Откроется окно Добавить реляционную запись.
5. В поле Имя введите **update_record**.
6. В области **Создать управляющие элементы** для выберите **Обновление существующей записи**.
7. Убедитесь, что выбрана опция **Добавить управляющие элементы ввода-вывода для отображения реляционной записи на web-странице**.
8. Нажмите кнопку **Далее**.
9. В поле Таблица разверните запись **W5SAMPLE** и выберите **ADS**.
10. Нажмите кнопку **Далее**. Откроется окно Добавить реляционную запись.

Фильтрация результатов

Реляционную запись можно связать только с одной записью из базы данных. По этой причине таблицу базы данных необходимо отфильтровать таким образом, чтобы в ней осталась только одна запись. (В предыдущем упражнении такое требование отсутствовало, поскольку при создании новой записи фильтрация результатов из базы данных не требуется.) Поскольку каждой записи базы данных присваивается уникальный ИД, в качестве критерия фильтрации применяется конкретный ИД.

1. На странице Добавить реляционную запись приведен список ссылок на задачи. В разделе **Задачи** выберите **Фильтрация результатов**. Откроется окно Фильтры, в столбце **Фильтр** которого показано

условие фильтрации по умолчанию ID = #{param.ID}. Окно Фильтры выглядит следующим образом:



Этот код фильтрует записи базы данных и отображает только запись с указанными ИД. Дополнительную информацию об этом условии вы получите в разделе Добавление действия строки позже в этом упражнении. Нажмите кнопку **Заккрыть**.

2. Нажмите кнопку **Далее**. Откроется страница Настроить управляющие элементы данных.
3. В области **Отображаемые поля** отмените выбор переключателей рядом со всеми полями, которые не следует показывать в форме ввода:
 - ИД
 - НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН
4. С помощью кнопок **Вверх** и **Вниз** расположите поля в следующем порядке:
 - ИД
 - НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН
5. Для поля ИД в выпадающем списке **Тип управляющего элемента** выберите **Поле вывода**. Пользователи могут просматривать ИД записи, однако не могут вносить в него изменения. Преобразование поля ИД в поле вывода позволяет избежать неполадок, связанных с совпадающими ИД.
6. Щелкните на пункте **Опции**. Откроется окно Опции. Убедитесь, что опция **Кнопка отправки** выбрана. В поле Метка введите **Обновление**. Нажмите **ОК**. Окно Добавить реляционную запись выглядит следующим образом:

Add Relational Record

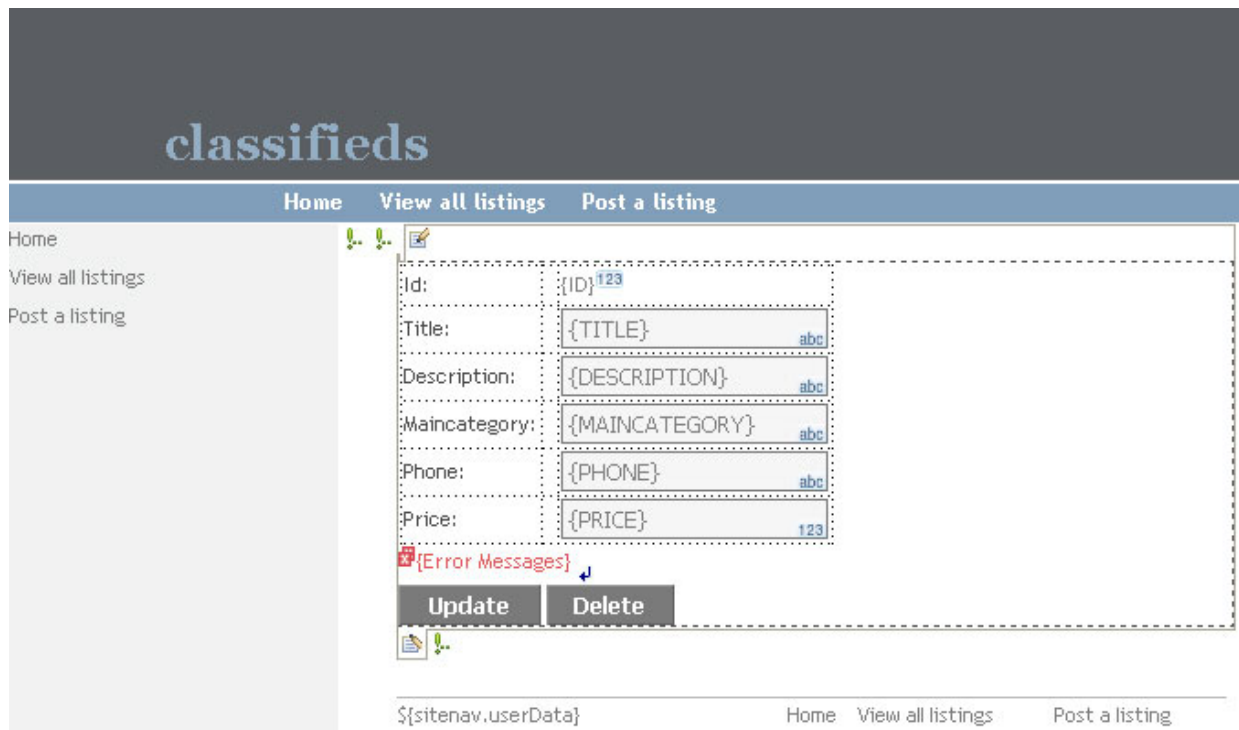
Configure Data Controls

Specify the columns to display and how to display them

Fields to display:

Field Name	Label	Control Type
☒ ID (java.lang.Integer)	Id:	Output Field
☒ TITLE (java.lang.String)	Title:	Input Field
☒ DESCRIPTION (java.lang.String)	Description:	Input Field
☒ MAINCATEGORY (java.lang.String)	Maincategory:	Input Field
☒ PRICE (java.math.BigDecimal)	Price:	Input Field
☒ PHONE (java.lang.String)	Phone:	Input Field
☐ TYPECODE (java.lang.String)	Typecode:	Input Field
☐ DATEOPEN (java.util.Date)	Dateopen:	Input Field
☐ CATEGORY_ID (java.lang.Integer)	Category_id:	Input Field
☐ TYPE (java.lang.String)	Type:	Input Field
☐ EMAIL (java.lang.String)	Email:	Input Field
☐ PHOTO (byte[])	Photo:	File Upload
☐ IMAGETYPE (java.lang.String)	Imagetype:	Input Field
☐ LOCATION (java.lang.String)	Location:	Input Field

7. Нажмите кнопку **Готово** для добавления на страницу формы обновления, как это показано ниже:



Программирование кнопок удаления и обновления

Снова следует добавить код, направляющий пользователя на страницу `all_records.jsp` для отображения списка записей с измененной записью.

1. Правой кнопкой мыши щелкните на созданной кнопке **Обновить** и выберите **Свойства**.
2. Нажмите кнопку **Добавить правило**. Откроется окно Добавить правило навигации.
3. В выпадающем списке Страница выберите **all_records.jsp**. В поле **Область применения правила** выберите опцию **Только это действие**. Убедитесь, что в этом поле указано значение `#{pc_Update_record.doUpdate_recordUpdateAction}`. Нажмите **ОК**.
4. Повторите шаги 1-3 для настройки кнопки Удалить. Убедитесь, что в поле **Область применения правила** указано значение `#{pc_Update_record.doUpdate_recordDeleteAction}`.
5. Сохраните страницу. Рекомендуется запустить на тестовом сервере файл `new_record.jsp`, чтобы добавить запись в базу данных и проверить возврат к странице `all_records.jsp`.

Добавить действие строки

Теперь вы создадите действие строки в таблицу на странице `all_records.jsp`, так что пользователь сможет выбрать запись базы данных для обновления. Параметр `#{param.ID}` представляет ИД обновляемой записи. Когда пользователь выбирает строку, ИД записи отправляется на страницу `update_record.jsp` в качестве параметра `#{param.ID}`. Затем в отфильтрованной реляционной записи, добавленной на страницу `update_record.jsp`, отображается только указанная запись.

1. На панели Меню проектов дважды щелкните на файле **all_records.jsp**. Файл будет открыт в редакторе. Щелкните по таблице с данными. Откройте панель Свойства.
2. В панели Свойства щелкните на вкладке **Действия строки** в разделе **hx:dataTableEx** в списке тегов HTML.
3. Выберите **Добавить действие, выполняемое в случае щелчка на строке**. Откроется окно Настроить действие строки. Выберите **Щелчок на строке отправляет запрос на сервер с информацией из строки в качестве параметров** и нажмите кнопку **ОК**. В таблицу данных добавляется новый столбец.
4. В панели Свойства выберите вкладку **hx:requestRowAction**.
5. Нажмите кнопку **Добавить правило**. Откроется окно Добавить правило навигации.
6. В выпадающем списке Страница выберите **update_record.jsp**. В поле **Область применения правила** выберите **Только это действие**. В поле **Только это действие** автоматически указывается значение

`{pc_All_records.doRowAction1Action}`, созданное при добавлении действия строки. Нажмите **ОК**. Если в поле не было автоматически указан этот метод, то введите его имя и добавьте следующее определение в ресурсе `ClassifiedsTutorial/Java: src/pagecode/All_records.java`:

```
public String doRowAction1Action() {  
    return "";  
}
```

7. В панели Свойства выберите вкладку **Параметр** в разделе `hx:requestRowAction`. Нажмите кнопку **Добавить параметр** и введите ИД в поле Имя. Параметр ИД необходимо связать со столбцом ИД. Щелчок на строке, параметр которой связан со столбцом ИД списка данных, позволяет задать параметр запроса в качестве ИД записи в списке.
 - a. Выберите **Значение** и нажмите кнопку **Выбрать объект данных страницы** .
 - b. На вкладке **Объект данных** разверните **all_recordlist (Service Data Object)** → **all_recordlist (ADS)** и выберите **select ID (java.lang.Integer)**.
 - c. Нажмите **ОК**.

Теперь щелкнув на строке, пользователь сможет обновить объявление. Сохраните файл и проверьте страницу. Обратите внимание, что сначала следует открывать страницу `all_records.jsp`, поскольку на ней расположена ссылка на страницу `update_record.jsp`.

Контрольные точки урока

Вы закончили урок 1.6. В этом уроке вы научились создавать Web-страницы, позволяющие обновлять записи базы данных.

Модуль 1: Итог

Вы закончили работу с модулем 1: Создание Web-страниц с соединениями с данными.

Пройденные уроки

Теперь вы знакомы со следующими задачами и процедурами:

- Отображение на Web-страницах информации из баз данных.
- Работа с реляционными записями, списками реляционных записей и таблицами данных.
- Отображение, создание, изменение и удаление записей базы данных с помощью Web-страницы.
- Связывание записи или набора записей с помощью действия строки в таблице данных.

В этом модуле приведен обзор технологии и инструментов `JavaServer Faces`. Однако возможности по улучшению внешнего вида, эффективности и удобства работы с Web-сайтом гораздо шире. В следующем модуле “Модуль 2: Добавление дополнительных функций” на стр. 26 будет показано, каким образом на основе сайта объявлений можно создать более привлекательное и сложное приложение. Продолжив обучение, вы сможете создавать сайты, которые выглядят следующим образом:

Home

View all listings »

Post a listing

View all listings

Title	Details	Category
W/V Passat wagon 	Description: 2000 good condition Price: \$1,000.00 Phone: 416-512-8665	Automotive
Magic Flute 	Description: Imagine it. Price: \$101.00 Phone: 314-159-265	Entertainment
Video Camera 	Description: Great camera, suitable for amateur or professional use. Price: \$900.00 Phone: 534-333-454	Entertainment
VCR 	Description: Old and worn but still working ok, hence low price. Price: \$40.00 Phone: 322-654-044	Entertainment
Flat Screen TV 	Description: Very clear picture, sound excellent too. Reluctant sale. Price: \$700.00 Phone: 456-456-944	Entertainment
1 2 3 4 5		

Модуль 2: Добавление дополнительных функций

В этом модуле вы научитесь пользоваться дополнительными функциями для изменения формата Web-страниц, добавлять функцию, позволяющую пользователям загружать файлы в базу данных, настраивать правила, автоматически направляющие пользователей к нужной странице, и автоматизировать выполнение некоторых административных задач, например генерации ключей.

Цели обучения

В данном модуле описаны расширенные способы использования данных из базы данных. В этом модуле рассмотрены следующие процедуры:

- Форматировать записи базы данных на Web-странице
- Добавление компонента, позволяющего загружать файлы в базу данных с Web-страницы.
- Автоматически переходить с одной страницы на другую
- Автоматизировать выполнение некоторых задач администрирования базы данных

Необходимое время

Для выполнения данного модуля потребуется примерно 1 час 30 минут. Изучение прочих фасетов динамических Web-сайтов может потребовать больше времени.

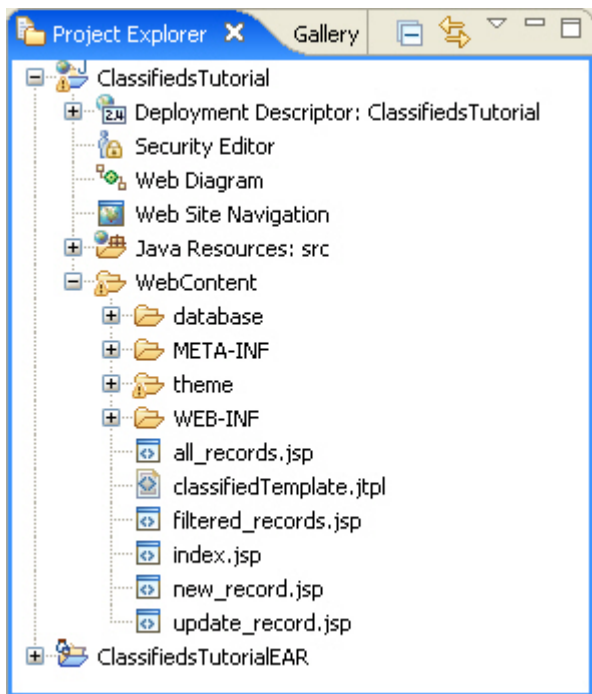
Предварительные требования

Если вы уже завершили работу с модулем 1: Создание Web-страниц с соединениями обмена данными, пропустите раздел предварительных требований и перейдите непосредственно к разделу “Урок 2.1: Форматирование таблицы данных” на стр. 30.

Если вы начинаете изучение учебника с модуля 2, предварительно необходимо импортировать требуемые ресурсы, задать целевой сервер и настроить соединение с базой данных.

Для импорта требуемых ресурсов выполните следующие действия:

1. Импортировать проект. Откройте проекцию Web (выберите **Окно → Открыть проекцию → Web**).
2. В проекции Web откройте панель Меню проектов и убедитесь, что проект ClassifiedsTutorial выглядит аналогично следующему рисунку:



Для задания целевого сервера выполните следующие действия:

1. В панели Меню ресурсов проекции Web правой кнопкой мыши щелкните на проекте **ClassifiedsTutorial** и выберите **Свойства**.
2. В списке свойств выберите **Сервер**.
3. В списке Сервер по умолчанию выберите сервер для применения по умолчанию. Нажмите кнопку **Применить**.
4. В списке свойств выберите **Целевые среды выполнения**.
5. В списке Среды выполнения щелкните на среде выполнения, соответствующей выбранному серверу. Нажмите **ОК**.

Если нужный сервер отсутствует в списке целевых сред выполнения, закройте рабочую среду и установите сервер. После установки сервера выполните инструкции по заданию целевого сервера.

Примечание: Если после установки целевых сред выполнения список Сервер по умолчанию пуст, может потребоваться дополнительная настройка сервера. Для настройки сервера выполните следующие действия:

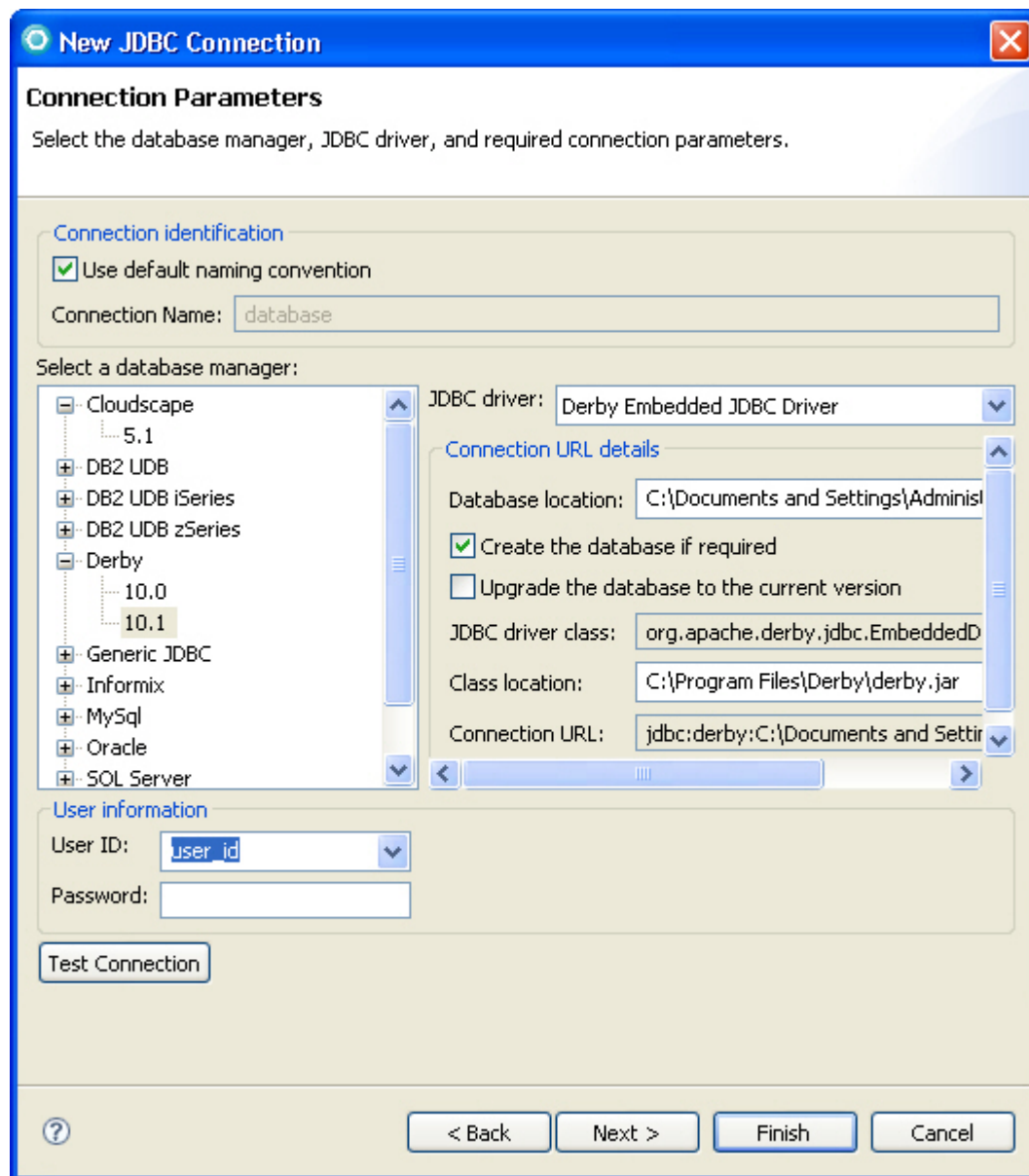
1. Правой кнопкой мыши щелкните на файле **all_records.jsp**, выберите **Выполнить как** → **Выполнить на сервере**.
2. Выберите **Задать сервер вручную**.
3. Выберите установленный сервер.
4. Настройте сервер, следуя инструкциям мастера. При первом запуске сервера может быть показано сообщение об ошибке. Для исправления ошибки задайте целевой сервер, как это указано выше, перезапустите сервер с помощью панели Серверы, а затем перезагрузите Web-страницу в браузере.

Вернувшись к рассмотренной выше процедуре задания целевого сервера, вы обнаружите, что настроенный сервер выбран в качестве сервера по умолчанию. Если сервер установлен, но не настроен, он не отображается в списке серверов.

Информация о разворачивании SDO на серверах, отличных от WebSphere Application Servers, приведена в разделе справки Разворачивание SDO на серверах, отличных от WAS.

Для настройки соединения с базой данных выполните следующие действия:

1. В панели Структура проекта правой кнопкой мыши щелкните на проекте **ClassifiedsTutorial** и выберите **Свойства**. Откроется окно Свойства ClassifiedsTutorial.
2. Выберите Соединения JDBC.
3. В панели Соединение JDBC нажмите кнопку **Создать**. Откроется окно Создать соединение JDBC.
4. Выберите **Создать новое соединение** и нажмите кнопку **Далее**.
5. В списке **Выбрать администратора базы данных** разверните запись **Derby** и выберите **10.1**.
6. В поле Расположение базы данных нажмите кнопку **Обзор** и выберите *<расположение-рабочей-области>* \ClassifiedsTutorial\WebContent\database, где *<расположение-рабочей-области>* - это каталог текущей рабочей области. Нажмите кнопку **ОК**, чтобы закрыть окно Выбрать папку.
7. В поле **Расположение класса** введите расположение одного или нескольких файлов JAR с реализацией класса драйвера JDBC, либо нажмите кнопку **Обзор**, чтобы выбрать нужное расположение в файловой системе. После выбора расположения файлов JAR нажмите кнопку **Открыть**.
8. Для обращения к базе данных может потребоваться указать ИД пользователя. Пароль не требуется. Мастер Создать соединение с базой данных выглядит следующим образом:



9. В мастере Создать соединение JDBC нажмите кнопку **Готово**. Вы вернетесь к окну Свойства ClassifiedsTutorial.
10. Выберите опцию **Настроить соединения проекта с базой данных**. Будет показано окно свойств соединений JDBC.
11. Выберите опцию **Изменить** рядом с разделом сведений о рабочих соединениях. В качестве типа соединения выберите **Использовать соединение диспетчера драйверов**. Нажмите кнопку **Готово**, а затем - кнопку **ОК**.

Вы можете просмотреть файлы Web-проекта учебника. Чтобы открыть файл дважды щелкните на нем на панели Проводник проектов. Для просмотра графического представления связей между страницами дважды щелкните на записи Навигация по Web-сайту в панели Меню проектов.

В этом примере вы, в основном, будете работать со следующими файлами:

all_records.jsp

Домашняя страница Web-сайта. На ней отображаются все объявления из базы данных.

new_record.jsp

Страница, позволяющая создать новое объявление.

update_record.jsp

Страница, позволяющая изменить или удалить объявление из базы данных.

classifiedTemplate.jtpl

Шаблон страниц сайта. Он содержит такие элементы, как таблица и серый баннер "Добро пожаловать в службу объявлений", отображаемый на каждой странице. Кроме того, эта страница содержит две навигационные вкладки под баннером, позволяющие перейти к домашней странице и к странице добавления нового объявления.

Урок 2.1: Форматирование таблицы данных

Таблица данных на странице `all_records.jsp` имеет утилитарный внешний вид. На ней отображаются все записи из базы данных, а также ссылки на другие страницы, однако выглядит это не очень привлекательно. В этом упражнении недостатки внешнего вида таблицы данных будут устранены за счет добавления таких элементов, как функция разбиения на страницы, правила стилей и рисунки продаваемых товаров.

Переименование заголовков столбцов

Зачастую названия столбцов в таблице с данными не должно совпадать с именами столбцов базы данных. В следующих шагах описывается как переименовать столбцы таблицы данных.

1. Дважды щелкните на странице **all_records.jsp** на панели Меню проектов
2. Щелкните на заголовке **Главная категория** в таблице данных.
3. На панели Свойства перейдите на вкладку **h:outputText** и вместо значения Главная категория в текстовом поле **Значение** укажите Категория.
4. При необходимости измените имена других столбцов.
5. Сохраните страницу.

Форматирование компонентов вывода

Вы можете отформатировать компоненты вывода. В следующих шагах мы отформатируем компонент вывода {ЦЕНА} для отображения в виде денежной единицы, а не обычного числа.

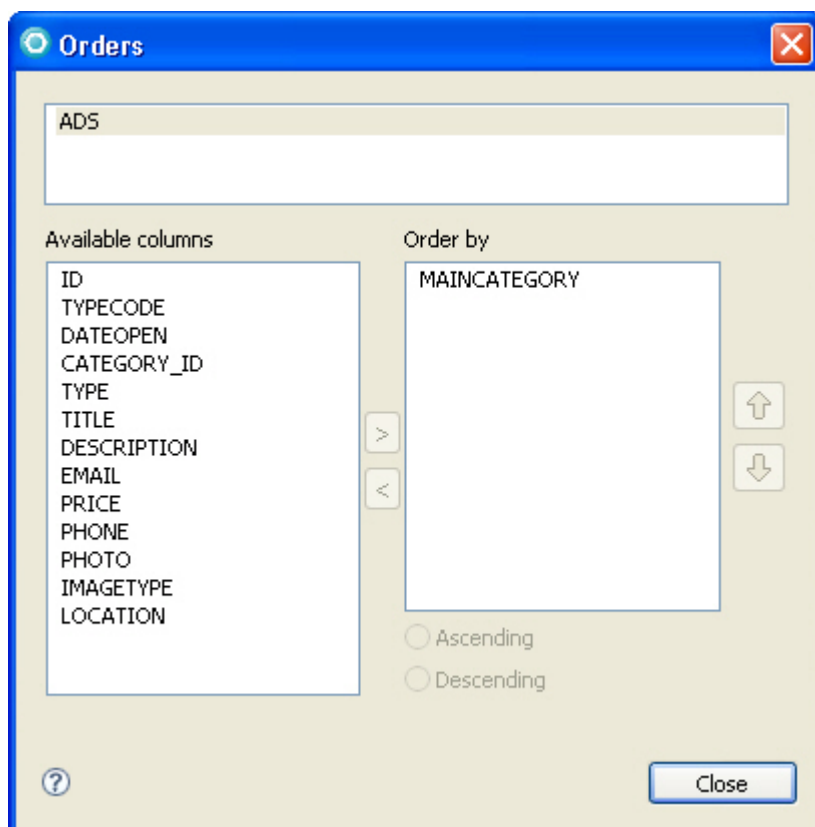
1. Щелкните на компоненте вывода {ЦЕНА} на странице.
2. На панели Свойства в выпадающем списке **Тип** выберите Денежная единица. Теперь цена всех элементов будет отображена как денежная единица, а не обычное число. Для прочих типов данных доступны и другие варианты - проценты, дата и время. Можно также настроить формат компонента вывода для отображения типа данных, не перечисленных на панели Свойства, например, телефонный номер.
3. Сохраните страницу.

Сортировка данных

Сейчас данные показываются беспорядочно. Чтобы создать более организованный список объявлений, упорядочьте его по категории, группировав схожие элементы.

1. В панели Данные страницы щелкните правой кнопкой мыши на записи **all_recordlist (ADS) (Окно → Показать панель → Данные страницы)** и выберите в контекстном меню пункт **Настроить**. Откроется окно Настроить список реляционных записей `window opens`. Появление предупреждения с сообщением о невозможности установить соединение с базой данных означает, что вы оставили сервер работающим после тестирования Web-сайта. В таком случае нажмите кнопку отмена во всех окнах и остановите сервер в соответствии с инструкциями, приведенными в разделе "Остановка сервера" упражнения Тестирование Web-сайта.
2. Выберите переключатели **Загрузить существующую запись или список записей из области** и **Повторно использовать определение метаданных из существующей записи или списка записей** и нажмите кнопку **Далее**.
3. В следующем окне выберите ADS и нажмите кнопку **Далее**.
4. В правой части следующего окна нажмите кнопку **Упорядочить результаты**. Откроется окно Порядок.

5. На панели Доступные столбцы выберите ГЛАВНАЯ КАТЕГОРИЯ.
6. Нажмите кнопку > для перемещения столбца ГЛАВНАЯ КАТЕГОРИЯ в область Упорядочить по панели. Это окно должно выглядеть следующим образом:



7. Нажмите кнопку **Заккрыть**, чтобы закрыть окно Порядок.
8. Нажмите кнопку **Готово** для применения изменений. Теперь данные будут рассортированы по категории. Сортировку в базе данных можно выполнить по любому столбцу.

Добавление функции разбиения на страницы

Вместо отображения каждой записи по одной в таблице с данными, можно использовать разбиение на страницы. При этом записи автоматически разбиваются на страницы заданного размера без создания новых файлов JSP в проекте.

1. Щелкните по таблице с данными.
2. В панели Свойства выберите **h:dataTableEx** в списке тегов HTML слева.
3. Перейдите на вкладку **Опции отображения** в области **h:dataTableEx**.
4. В поле **Число строк на странице** введите 5.
5. Нажмите кнопку **Добавить разбиение на страницы**. Внизу таблицы с данными появится пример внешнего вида разбиения на страницы. Визуализация разбиения на страницы в проектировщике страниц является заменителем и не отражает фактический вид, так как он будет виден только при загрузке страницы в браузере.
6. Сохраните страницу. Вы можете поэкспериментировать с разными типами разбиения на страницы и компонента статистики страниц для выбора наиболее подходящего варианта для страницы.

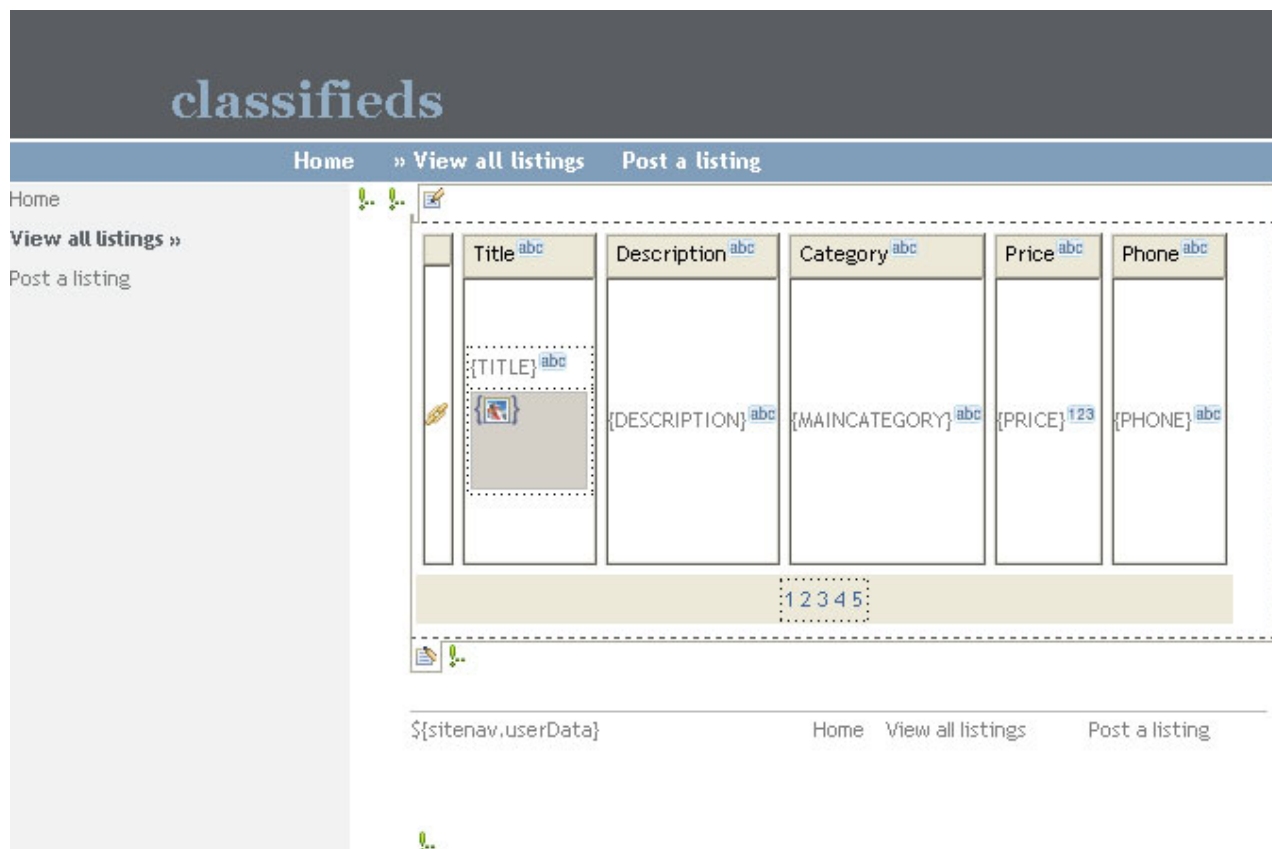
Расположение компонентов в группе элементов в виде списка

Вместо ровно одного компонента вывода в каждом столбце таблицы с данными можно сочетать элементы со столбцами для получения более привлекательной разметки. С помощью групп элементов можно

упорядочить компоненты, точно так же как и скрытые таблицы HTML. Например, добавив изображения для улучшения облика сайта вы сможете поместить большую часть текстовой информации из объявлений в один столбец и назвать его СВЕДЕНИЯ.

1. Откройте лоток Расширенные компоненты Faces в панели Палитра. Перенесите компонент **Панель - Группа элементов** на столбец ЗАГОЛОВОК таблицы данных. Откроется окно Выбрать тип.
2. В окне Выбрать тип в качестве типа группы элементов выберите **Поле** и нажмите кнопку **ОК**. Такая группа элементов размещает добавленные в нее компоненты в одном столбце или строке; при необходимости ее можно развернуть горизонтально или вертикально.
3. Щелкните на группе элементов чтобы выбрать ее.
4. На панели Свойства в поле **Ориентация** выберите **Вертикальная**.
5. Перенесите компонент **{НАЗВАНИЕ}** на группу элементов. После добавления компонента в группу элементов поясняющий текст исчезнет.
6. Перенесите компонент **Изображение** из лотка Расширенные компоненты Faces палитры на группу элементов. Рекомендуется поместить компонент изображения вниз группы дабы убедиться, что изображение будет размещено под названием.
7. Щелкните на добавленном компоненте изображения.
8. В области **Размер** панели Свойства укажите в полях **Ширина** и **Высота** значения 60 и 50 соответственно. Это обеспечит отображение изображений на странице размером 60x50 пикселей, вне зависимости от их истинного размера.
9. Привяжите компонент изображения к столбцу ФОТО all_recordlist, переместив столбец ФОТО из панели Данные страницы в компонент изображения. Это приведет к тому, что компонент изображения отобразит данные изображения, обнаруженные в столбце ФОТО для каждой записи. Если столбец ФОТО (ByteArray) не показан, в панели Данные страницы щелкните правой кнопкой мыши на записи **all_recordlist (ADS)** и выберите в контекстном меню пункт **Настроить**. При необходимости закройте окно URL, нажав кнопку **ОК**. В окне Настроить список реляционных записей нажмите кнопку **Далее**. Выберите ADS и нажмите кнопку **Далее**. Выберите переключатель рядом со столбцом **ФОТО (ByteArray)** и нажмите кнопку **Далее**.
10. Щелкните на добавленном рисунке. На вкладке **hx:graphicImageEx** панели Свойства нажмите кнопку **Выбрать объект данных страницы**  рядом с полем **Тип**. Откроется окно Выбрать объект данных страницы.
11. Разверните **all_recordlist (Service Data Object)** → **all_recordlist (ADS)**.
12. Выберите **IMAGETYPE (String)**.
13. Нажмите кнопку **ОК** и сохраните страницу.

Теперь страница должна выглядеть следующим образом:

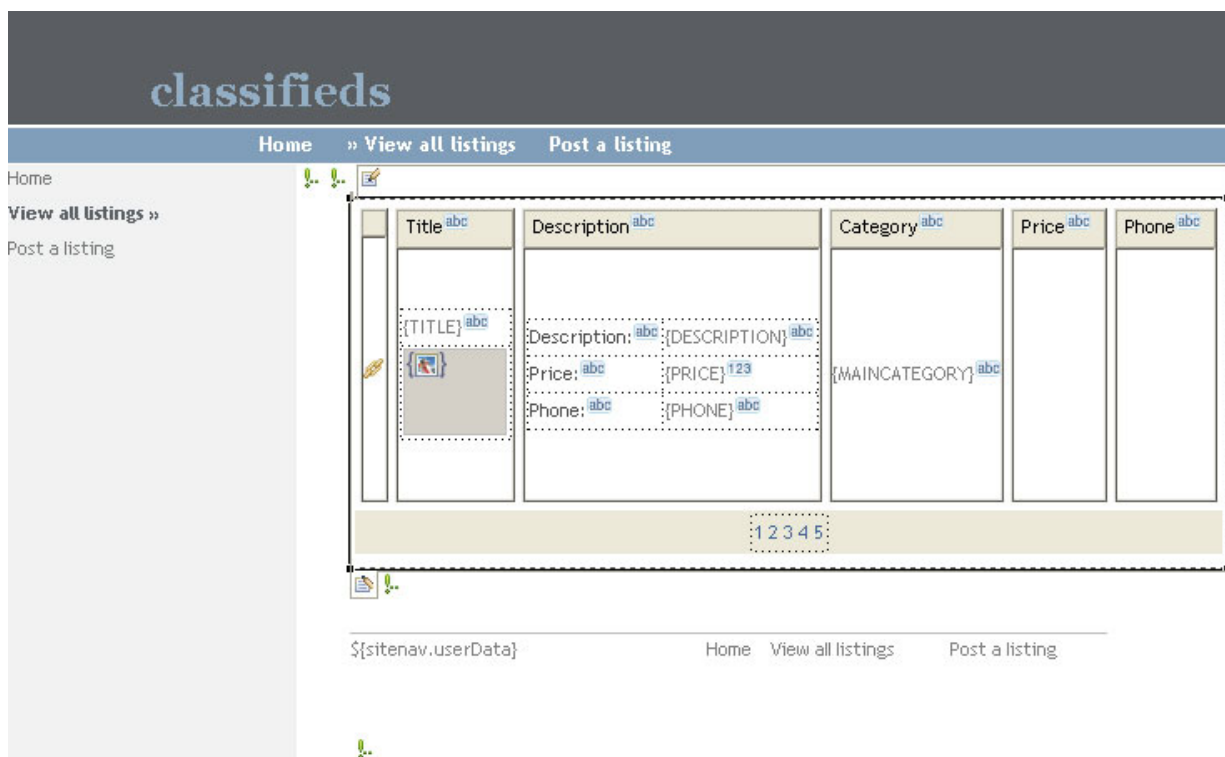


Уточнение разметки с помощью сетки

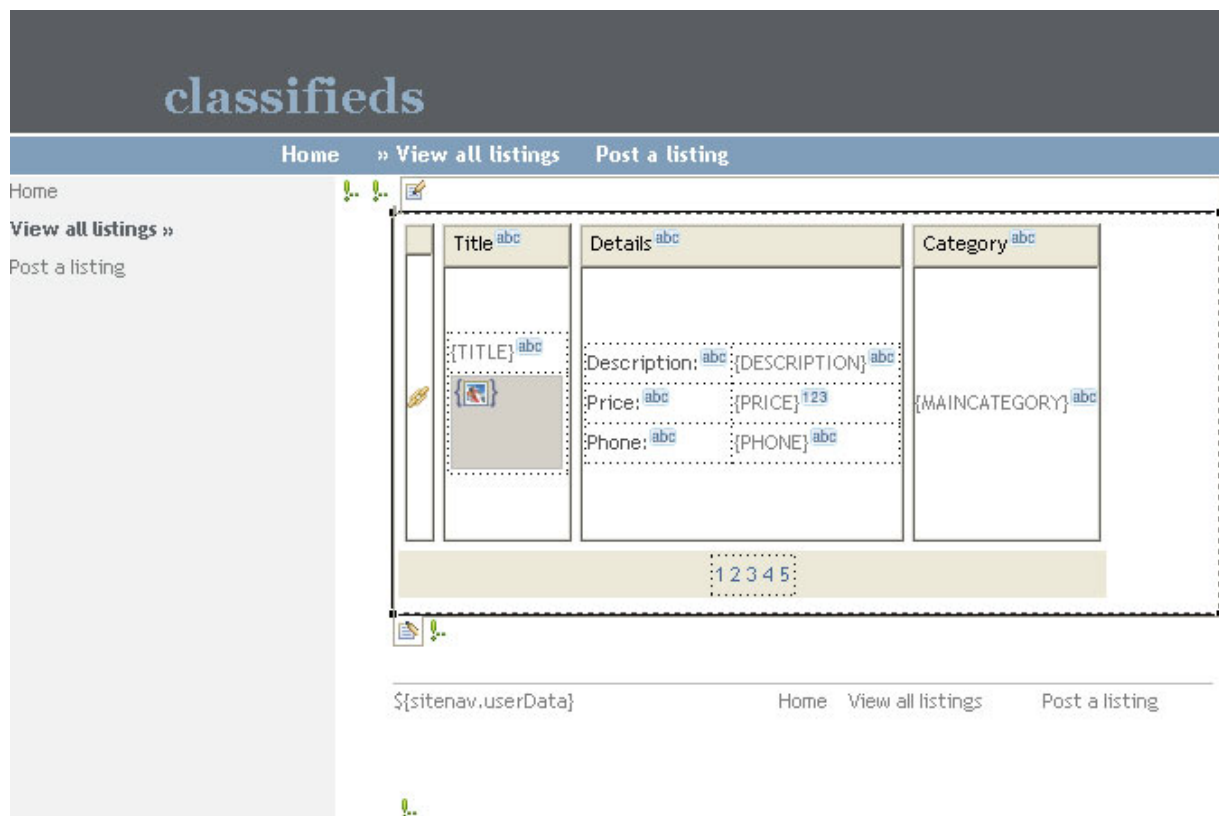
Группа элементов позволяет организовать компоненты наподобие таблицы. Группа элементов типа "список" может содержать одну строку или столбец, но группа элементов типа "сетка" может содержать любое число строк и столбцов. В описанных ниже шагах мы переместим компоненты {ЦЕНА} и {ТЕЛЕФОН} в столбец ОПИСАНИЕ таблицы данных и добавим метки для них.

1. С помощью указателя мыши перенесите компонент **Панель - Группа элементов** из лотка Расширенные компоненты Faces палитры на столбец ОПИСАНИЕ таблицы данных. Откроется окно Выбрать тип.
2. Выберите **Сетка** в качестве типа добавляемого компонента и нажмите кнопку **ОК**.
3. Щелкните на новой группе элементов "сетка".
4. На панели Свойства укажите для числа столбцов значение 2.
5. Перенесите компонент **Вывод** из лотка Компоненты Faces палитры на сетку. Этот компонент вывода станет меткой для описания продаваемого элемента. У каждого компонента вывода будет метка вроде этой, расположенная в левой ячейке таблицы.
6. Щелкните на компоненте **Вывод** (в проектировщике страниц он отображается в качестве **outputText**) и с помощью панели Свойства в поле **Значение** укажите Описание:
7. Перенесите существующий компонент {ОПИСАНИЕ} к нижнему краю сетки. Если при перемещении компонентов в группе элементов у вас возникли затруднения, попробуйте нажать кнопку мыши и посмотреть на расположение курсора. Оно указывает местоположение компонента после перемещения. Для данного упражнения необходимо отпустить кнопку мыши когда курсор будет находиться справа от предыдущего компонента.
8. Перенесите другой компонент **Вывода** из палитры и положите его справа от группы элементов.
9. На панели свойства в поле **Значение** укажите Цена:.
10. Переместите существующий компонент вывода {ЦЕНА} вправо от текста вывода цены.
11. Аналогичным образом перенесите новый компонент **Вывод** {ТЕЛЕФОН} и укажите для него метку Телефон:.

12. Переместите существующий компонент {ТЕЛЕФОН} в группу элементов "сетка" справа от текста вывода телефона. Теперь страница должна выглядеть следующим образом:



13. Удалите пустые столбцы ЦЕНА, ТЕЛЕФОН из таблицы данных. Для удаления столбца щелкните на нем и откройте панель Свойства. Затем выберите **h:dataTable** в списке тегов HTML в левой части панели и выберите столбец для удаления из списка в правой части панели, затем нажмите кнопку Удалить.
14. Щелкните на заголовке в столбце Описание и на панели Свойства в поле **Значение** укажите Сведения. Теперь страница должна выглядеть следующим образом:



15. Сохраните страницу и протестируйте ее. Если ранее вы не работали с приложениями на сервере, обратитесь к раздел “Урок 1.3: Тестирование Web-сайта” на стр. 13.

В ходе тестирования страница должна выглядеть следующим образом:

classifieds

Home » View all listings Post a listing

Home

View all listings »

Post a listing

View all listings

Title	Details	Category
WV Passat wagon 	Description: 2000 good condition Price: \$1,000.00 Phone: 416-512-8665	Automotive
Magic Flute 	Description: Imagine it. Price: \$101.00 Phone: 314-159-265	Entertainment
Video Camera 	Description: Great camera, suitable for amateur or professional use. Price: \$900.00 Phone: 534-333-454	Entertainment
VCR 	Description: Old and worn but still working ok, hence low price. Price: \$40.00 Phone: 322-654-044	Entertainment
Flat Screen TV 	Description: Very clear picture, sound excellent too. Reluctant sale. Price: \$700.00 Phone: 456-456-944	Entertainment

1 2 3 4 5

Объявления, не связанные с изображениями, отображаются в качестве неработающих ссылок. В следующем уроке рассмотрена процедура добавления на страницу компонента загрузки файлов, позволяющего добавлять и изменять изображения объявлений.

Контрольные точки урока

Вы закончили урок 2.1. В этом уроке вы ознакомились с различными способами форматирования страниц.

Теперь вы знакомы со следующими задачами и процедурами:

- Изменение заголовков столбцов:
- Сортировка данных в таблице.
- Разбиение данных на несколько Web-страниц.
- Изменение разметки таблиц на Web-страницах с помощью групп элементов.

Урок 2.2: Применение компонента загрузки файлов

В этом уроке показано, каким образом пользователям можно разрешить загружать рисунки для объявлений. Особенно важна эта функция для страниц `new_record.jsp` и `update_record.jsp`. Компонент загрузки файлов позволяет пользователям просматривать файловые системы, загружать файлы в базу данных и немедленно отображать эти файлы в базе данных.

Так как процедура использования компонента загрузки файлов схожа для процедуры создания и обновления объявлений, данная процедура подойдет для обновления страницы с учетом текущего изображения. При необходимости можно внести некоторые изменения в страницу создания новой записи после обновления страницы.

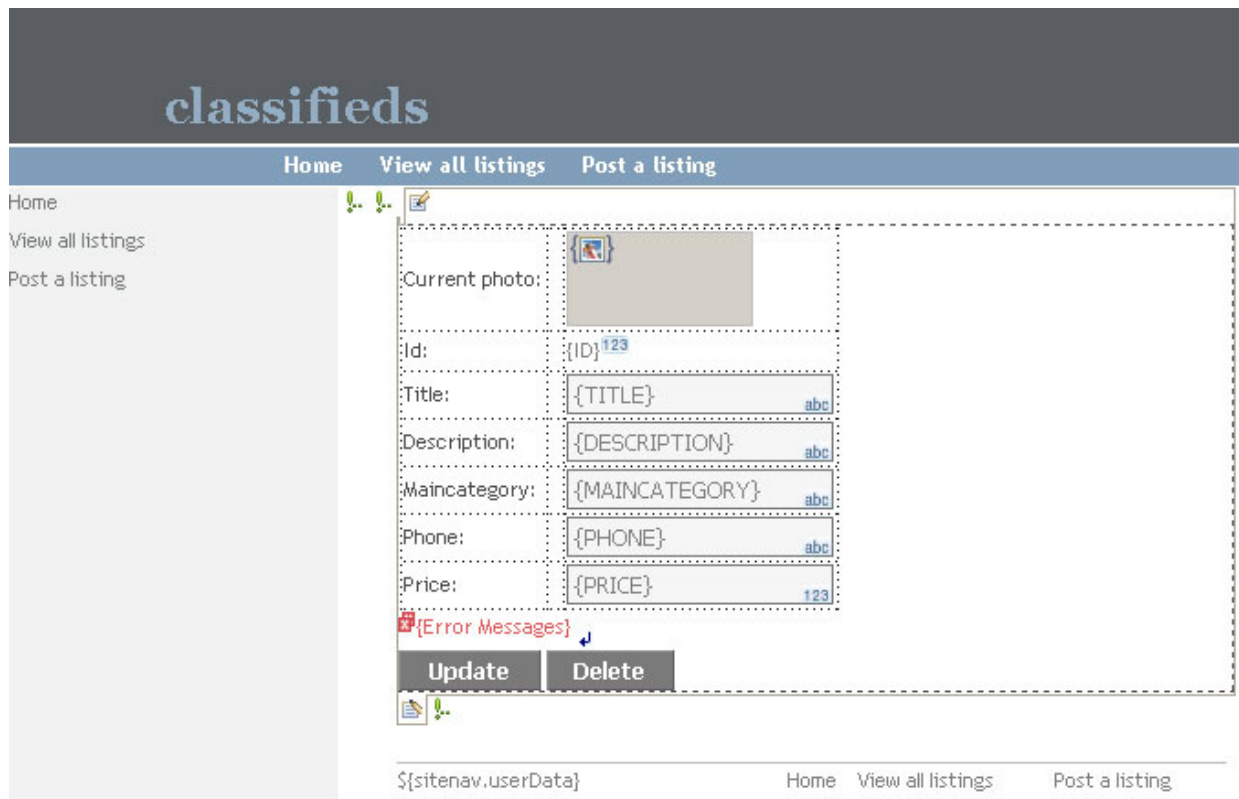
Добавление фотографии на страницу

Вы заметите, что форма обновления является обычной таблицей HTML со статичным текстом и некоторыми компонентами ввода, привязанными к столбцам в реляционной записи `update_record`. Учитывая это можно добавить строки и столбцы для изменения формы, точно так же, как и при работе с любой другой таблицей HTML. В последующих шагах вы добавите новую строку для отображения текущего фото.

Добавление строк и столбцов более сложно для таблицы с данными, отображающей список реляционных записей. Информация о добавлении столбцов к таблице с данными приведена в разделе “Урок 1.2:

Подключение к базе данных и отображение данных на Web-странице” на стр. 7.

1. Дважды щелкните на странице **update_record.jsp** в панели Меню проектов.
2. Поместите курсор в первую (верхнюю левую) ячейку таблицы. Эта ячейка помечена как ИД.
3. В главном меню выберите **Таблица → Указать и добавить**. Откроется окно Добавить строки и столбцы. Выберите **Строки**, затем - **Над выбранной ячейкой**. Будет создана новая строка сверху таблицы, которая будет содержать текущее фото для записи. Нажмите **ОК**.
4. В крайней левой ячейке новой строки введите метку Текущая фотография:.
5. Переместите компонент **Изображение** из лотка Компоненты Faces палитры в самую крайнюю ячейку новой строки. В отличие от предыдущего упражнения, вы позволите пользователю задать полноразмерное изображение без ограничения размеров, поэтому не изменяйте значения ширины и высоты на панели Свойства.
6. Привяжите компонент изображения к столбцу ФОТО для `update_record`, переместив столбец ФОТО из панели Данные страницы в новый компонент изображения. Компонент изображения теперь привязан к столбцу ФОТО базы данных.
7. Не отменяя выбора изображения, на панели Свойства нажмите кнопку Выбрать объект данных страницы рядом с полем Тип. Откроется окно Выбрать объект данных страницы.
8. Выберите **update_record (Service Data Object) → update_record (ADS) → IMAGETYPE (String)**.
9. Нажмите кнопку **ОК**. На странице будет показана текущая фотография для объявления. Страница должна выглядеть следующим образом:



Добавление компонента загрузки файлов на страницу

Затем добавьте новую строку внизу таблицы для хранения компонента загрузки файлов.

1. Поместите курсор в последней строке, щелкнув на нем.
2. В строке меню выберите **Таблица** → **Добавить строку снизу**.
3. В первой ячейке новой строки введите в метку Новая фотография:.
4. Переместите компонент **Загрузка файла** из лотка Компоненты Faces палитры в последнюю ячейку новой строки.
5. Также, как и в случае с компонентом изображения, свяжите компонент загрузки файлов к столбцу **ФОТО** реляционной записи `update_record`. В текстовом поле будет показана строка `{ФОТО}`, которая указывает, что этот компонент связан со столбцом **ФОТО** и в этот столбец будет добавлен загруженный файл. Страница должна выглядеть следующим образом:

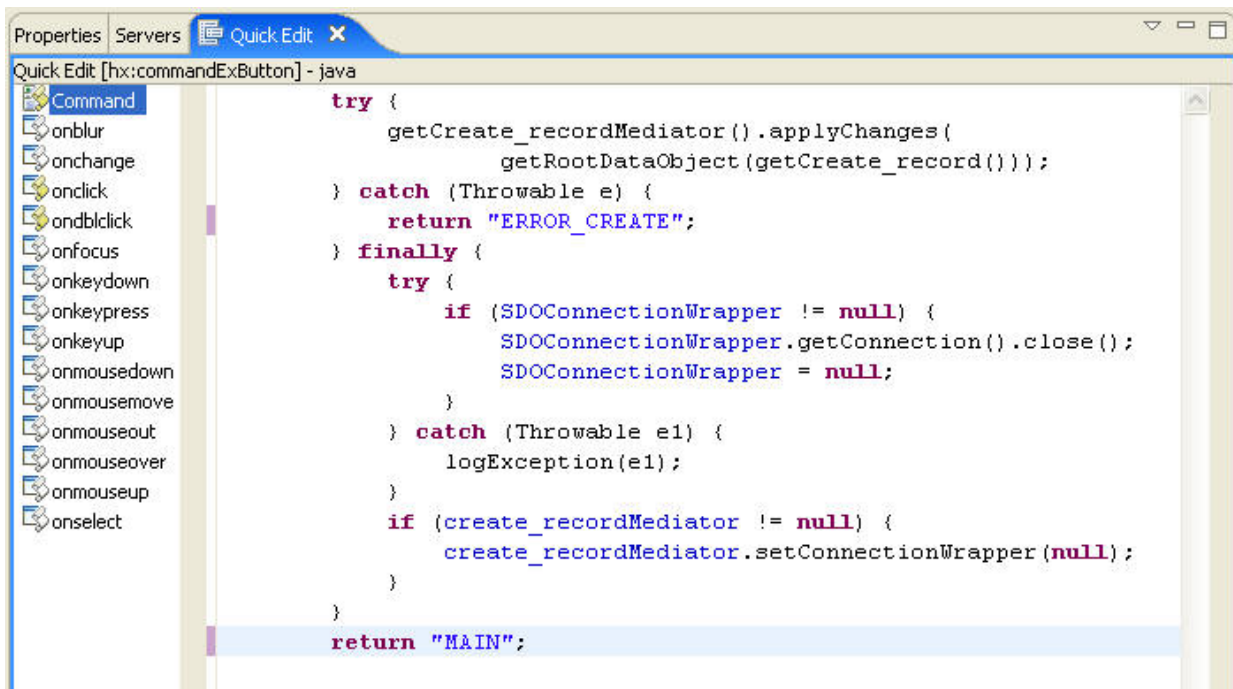
Настройка правил

1. Дважды щелкните на странице **new_record.jsp** в панели Меню проектов.
2. Нажмите кнопку **Добавить новый список**.
3. На панели Свойства нажмите кнопку **Добавить правило**. Откроется окно Добавить правило навигации. Первое правило отправляет пользователю страницу ошибки **create_error.jsp** в случае неполадок в ходе создания новой записи или ее фиксации в базе данных.
4. В поле **Страница** введите **error_create.jsp**. В этом учебнике предполагается, что эта страница существует.
5. Выберите переключатель **Имя варианта**.
6. Введите **ERROR_CREATE** в текстовом поле рядом с переключателем **Имя варианта**.
7. Выберите переключатель **Только данная страница**, поскольку указанная ошибка может возникнуть только на этой странице Web-сайта. Нажмите **ОК**. Следующее правило применяется для перехода на страницу **all_records.jsp**, если указан допустимый ИД.
8. Нажмите кнопку **Добавить правило**. Снова откроется окно Добавить правило навигации.
9. В выпадающем списке **Страница** выберите **all_records.jsp**.
10. Выберите переключатель **Имя варианта** и в текстовом поле рядом с ним введите **MAIN**.
11. Поскольку это правило будет использоваться и на других страницах (например, **update_record.jsp**), в разделе **Область применения правила** выберите переключатель **Все страницы**. Нажмите **ОК**. Оба правила отображаются в панели Свойства.

Возврат псевдонимов из действия кнопки

Теперь следует настроить применение правил навигации. Для этого в код кнопки **Добавить новый список** добавляются два оператора **return**. Эти операторы вызывают подходящий псевдоним для отображения пользователю страницы в соответствии с правилом навигации.

1. Нажмите кнопку **Добавить новый список**.
 2. Откройте окно **Быстрое редактирование**.
 3. В панели **Быстрое редактирование** найдите строку `} catch (Throwable e) {`. Эта функция **catch** вызывается в случае ввода совпадающего ИД.
 4. Удалите весь текст между открывающей фигурной скобкой `{` в конце этой строки и следующей закрывающей фигурной скобкой `}`. Фигурные скобки удалять не следует.
 5. Между фигурными скобками вставьте следующий текст: `return "ERROR_CREATE";`
 6. Затем удалите весь текст после последней закрывающей фигурной скобки `}`. Вместо него введите следующий текст: `return "MAIN";`
- Это удалит действие `gotoPage`, добавленное вами при выполнении упражнения 1.4. Вам больше не нужен этот код, поскольку правила навигации делают то же самое.
- Теперь код кнопки должен выглядеть следующим образом:



7. Сохраните страницу и протестируйте ее при необходимости.

(Необязательно) Вы можете создать простую страницу `create_error.jsp` с описанием ошибки при создании списка и инструкциями по ее исправлению. Затем вы сможете проверить правила навигации, добавив новый список с совпадающим ИД (например, 1).

Контрольные точки урока

Вы закончили урок 2.3. В этом уроке вы научились применять правила навигации, создающие ошибку, если пользователь пытается ввести недопустимый ИД, или направляющие пользователя на страницу `all_recordlist.jsp`, если пользователь вводит допустимый ИД.

Урок 2.4: Применение функции автоматического создания ключей

В предыдущем уроке были настроены правила навигации, обеспечивающие ввод уникального ИД при добавлении объявления. Такой подход может вызвать значительные затруднения в работе пользователей сайта, поскольку уникальные ИД можно найти только экспериментальным путем. В данном уроке вы настроите автоматическое создание ключей, чтобы база данных автоматически присваивала уникальный номер каждой новой записи базы данных.

автоматическая генерация ключей является сложной темой, вкратце - база данных может выбирать новые ключи при наличии специальной таблицы, зарезервированной для генерирования ключей. В этой таблице должен быть список неиспользуемых ключей в одном столбце (столбец *приращения*) и возрастающий список чисел, начиная с 1, во втором столбце (столбце *идентификаторов*). Когда базе данных потребуется новый ключ, она извлечет его из строки с номером 1 в столбце идентификаторов и затем получит новый ключ, готовый для следующего раза.

Дополнительные сведения об автоматическом создании ключей.:

Перед применением функции автоматического создания ключей рекомендуется ознакомиться с некоторыми особенностями ее работы. Для применения автоматической генерации ключей необходимо, чтобы в базе данных существовала таблица, специально предназначенная для этой цели. Эта таблица состоит из двух столбцов:

- Столбец *приращения* содержит доступные ключи. Из этого столбца база данных извлекает уникальные ключи.
- Столбец *идентификаторов* представляет собой набор чисел с одним экземпляром единицы. В соответствии с этим столбцом база данных выбирает ключ из столбца приращения. Столбец идентификаторов является первичным ключом таблицы создания ключей.

В ходе выбора уникального ключа база данных находит строку со значением 1 в столбце идентификаторов. В столбце приращения в этой строке расположен следующий доступный ключ. База данных применяет этот ключ и обновляет таблицу, подготавливая следующий уникальный ключ.

Ниже приведен пример таблицы создания ключей. Какой ключ будет выбран базой данных? Ответ приведен после таблицы.

Таблица 1. Таблица создания ключей

Столбец идентификаторов	Столбец приращения
3	78
4	3
1	65
2	12

Следующий доступный ключ - 65, поскольку в той же строке столбца идентификаторов указано значение 1.

После выборки уникального ключа таблица обновляется для следующей операции создания ключа. При необходимости база данных может одновременно загрузить несколько ключей, выбрав несколько строк из столбца приращения.

Таким образом, для работы функции автоматического создания ключей необходима таблица, состоящая из двух столбцов - столбца первичного ключа (столбца идентификаторов), а также столбца для хранения следующего доступного ключа. Для инициализации такой таблицы достаточно одной записи, в которой в столбце идентификаторов будет указано значение 1, а в столбце приращения - первый доступный ключ. После настройки описанной таблицы функция автоматического создания ключей готова к работе.

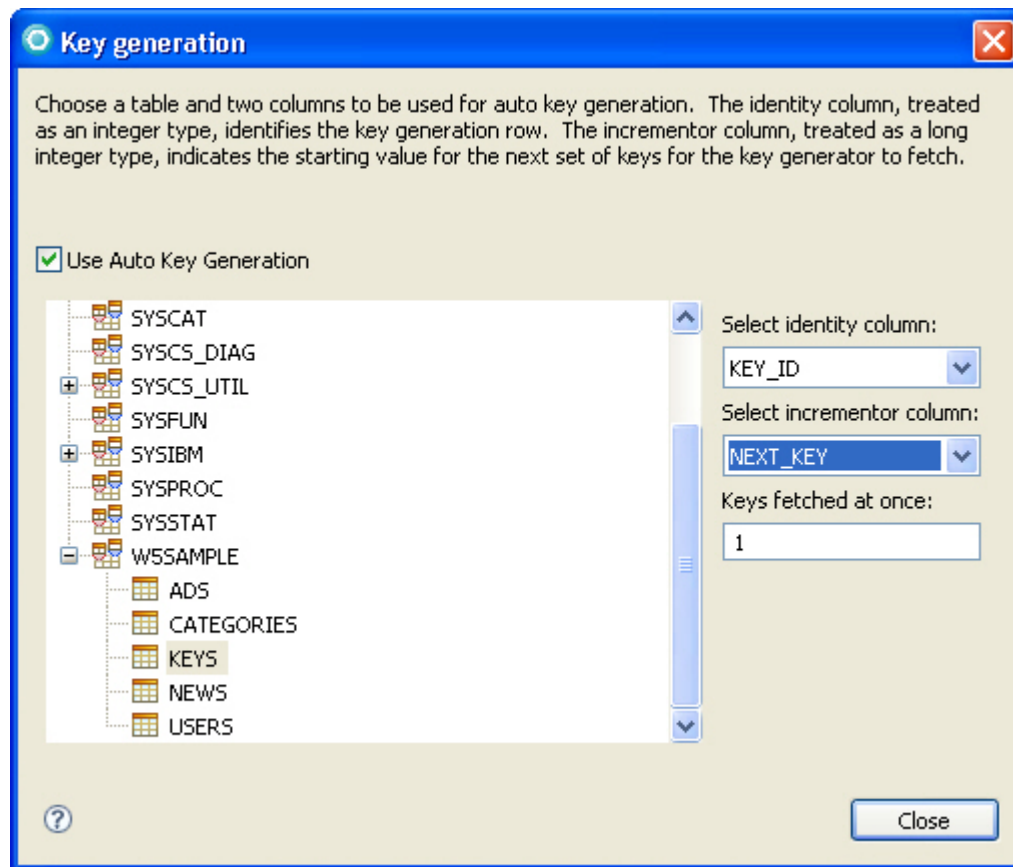
Настройка автоматического создания ключей

В примере базы данных для данного учебника имеется таблицы генерации ключей с именем KEYS. В двух ее столбцах, как описано выше, содержатся новые ИД для всех записей. В данной процедуре вы настроите реляционную запись `create_record` для извлечения ИД из таблицы KEYS.

1. Дважды щелкните на странице `new_record.jsp` в панели Меню проектов.
2. Дважды щелкните на реляционной записи `create_record` в панели Данные страницы. Появится окно Настроить реляционную запись.
3. Выберите переключатель **Повторно использовать определение метаданных из существующей записи или списка записей** и **Заполнять запись существующими данными из базы данных**, а затем нажмите кнопку **Далее**.
4. Выберите соединение с базой данных в выпадающем списке и нажмите кнопку **Далее**.
5. В правой части окна Настроить реляционную запись отображается список ссылок. Выберите **Автоматическое создание ключей**. Откроется окно Создание ключей.
6. Выберите **Использовать автоматическое создание ключей**.
7. Разверните список W5SAMPLE и выберите таблицу **KEYS**.
8. В поле **Выбрать столбец идентификаторов** выберите **KEY_ID**.
9. В поле **Выбрать столбец приращения** выберите **NEXT_KEY**.

Так как для ИД объявления вам понадобится только один ключ, оставьте параметр "Число извлекаемых за раз ключей" равным 1. Если вам понадобится несколько ключей, при таком значении параметра база данных будет выбирать их все за раз.

Окно Настроить объект данных должно выглядеть следующим образом:



10. Закройте окно Создание ключей с помощью кнопки **Заккрыть**. Затем нажмите кнопку **Готово** для применения изменений. Теперь поле ИД будет создаваться автоматически для каждой новой записи. Теперь необходимо удалить поле ввода ИД, чтобы пользователь не смог вручную ввести значение.
11. Поместите курсор на верхней строке таблицы формы ввода, щелкнув на текст **ИД**.
12. Выберите **Таблица** → **Удалить** → **Удалить строку**.
13. Сохраните страницу.

При необходимости просмотра сгенерированного для вас ключа можно удалить компонент ввода для данного ИД и заменить его на компонент вывода, привязанный к столбцу ИД для `create_record`. Затем автоматически сгенерированный ключ отобразится вверху формы, но будет недоступен пользователю.

Проверка заверченного учебника

Когда ваше Web-приложение будет готово к опубликованию, вам потребуется сервер для хостинга, чтобы пользователи смогли обращаться к Web-сайту через Internet. Однако для тестирования Web-сайта можно воспользоваться доступной средой выполнения для имитации сервера. Инструкции по тестированию Web-сайта приведены в разделе “Урок 1.3: Тестирование Web-сайта” на стр. 13.

Контрольные точки урока

Вы закончили урок 2.4. В этом уроке вы научились настраивать функцию автоматического создания ключей.

Модуль 2: Итог

Вы завершили работу с модулем 2: Добавление дополнительных функций.

Пройденные уроки

Теперь вы знакомы со следующими задачами и процедурами:

- Форматировать записи базы данных на Web-странице
- Добавление компонента, позволяющего загружать файлы в базу данных с Web-страницы.
- Переходить с одной страницы на другую с помощью правил навигации
- Использовать функцию автоматического создания ключей

Дополнительные ресурсы

Ниже приведены ссылки на дополнительную информацию об этом продукте, а также о технологии JavaServer Faces.

Информация, связанная с данной



Технология JavaServer Faces



JSF в developerWorks

Отображение динамических сведений на Web-страницах с JavaServer Faces

Вы выполнили учебник Отображение динамических сведений на Web-страницах с JavaServer Faces. Из него вы узнали как использовать JavaServer Faces для создания Web-страниц, способных извлекать информацию из базы данных.

Пройденные уроки

Выполнив данный учебник, вы узнали о следующих задачах или концепциях:

- Подключение Web-страниц к базе данных.
- Создание страниц, обеспечивающих создание, изменение и удаление записей базы данных.
- Отправка данных между страницами.
- Форматирование динамического содержимого Web-страниц.
- Компоненты JavaServer Faces и JSF.

Дополнительные ресурсы

Для получения дополнительных сведений о темах, затронутых в данном учебнике, обратитесь к следующим источникам:

Информация, связанная с данной



Технология JavaServer Faces



JSF в developerWorks