



**JavaServer Faces を使用して動的情報を Web ページに
表示する**

目次

JavaServer Faces を使用して動的情報を Web ページに表示する 1

JavaServer Faces を使用して動的情報を Web ページ
に表示する 2

モジュール 1: データ接続を扱う Web ページの作成 4

演習 1.1: 必須リソースのインポートとターゲット
ト・サーバーの設定 5

演習 1.2: データベースへの接続と Web ページ上
でのデータの表示 7

演習 1.3: Web サイトのテスト 14

演習 1.4: 表示する新規レコードの作成とデータ
ベースの更新 16

演習 1.5: 「実行」ボタンのプログラミング . . . 20

演習 1.6: 更新ページの作成 23

モジュール 1: 要約 27

モジュール 2: 拡張機能の追加 28

演習 2.1: データ・テーブルのフォーマット . . . 32

演習 2.2: ファイル・アップロード・コンポーネ
ントの使用 39

演習 2.3: ナビゲーション規則の使用 41

演習 2.4: 自動キー生成の使用 43

モジュール 2: 要約 46

JavaServer Faces を使用して動的情報を Web ページ
に表示する 46

JavaServer Faces を使用して動的情報を Web ページに表示する

このチュートリアルでは、新聞の案内広告欄として機能する動的 Web サイトの設計方法を学習します。この Web サイトは、JavaServer Faces (JSF) コンポーネントとサービス・データ・オブジェクト (SDO) を使用する、完全な J2EE Web アプリケーションです。このアプリケーションは JSF テクノロジーを使用して、案内広告データを保管しているデータベースにリンクする動的 Web ページを作成します。JSF コンポーネントを対象のページにドラッグすることによって、CRUD アプリケーション (データベースからの作成、読み取り、更新および削除 (create, read, update, delete) が可能であるため、このように呼ばれます) である JSF Web アプリケーションを作成することができます。サイト・ユーザーは、これらの機能を使用して、データベース内の案内広告を管理できます。

学習目標

このチュートリアルは 2 つのモジュールに分割され、それぞれに学習目標があります。いずれか一方のモジュールを選択することも、あるいは両方のモジュールを選択することもできます。それぞれのモジュールで、Web サイトが適切に作動するように課題を完了してください。

モジュール 1: データ接続を扱う Web ページの作成

このモジュールでは、Web ページで、データベースへの接続をセットアップする方法、およびそのデータベースから情報を表示する方法を学習します。このモジュールでは、以下に挙げることを行います。

- Web ページをデータベースに接続する。
- データベースのレコードを表示、作成、編集、および削除するページを作成する。
- あるページから他のページにデータを送る

モジュール 2: 拡張機能の追加

このモジュールでは、データベースのデータを、より効果的に使用方法を学習します。このモジュールでは、以下に挙げることを行います。

- Web ページでデータベース・レコードをフォーマットする
- Web ページからデータベースへのファイルのアップロードを可能にするコンポーネントを追加する
- 自動的にページ間をナビゲートする
- データベース管理タスクの一部を自動化する

準備が完了したならば、『モジュール 1: データ接続を扱う Web ページの作成』を始めてください。

所要時間

合計: 3 時間 30 分

モジュール 1: 2 時間

モジュール 2: 1 時間 30 分

JavaServer Faces を使用して動的情報を Web ページに表示する

このチュートリアルでは、新聞の案内広告欄として機能する動的 Web サイトの設計方法を学習します。この Web サイトは、JavaServer Faces (JSF) コンポーネントとサービス・データ・オブジェクト (SDO) を使用する、完全な J2EE Web アプリケーションです。このアプリケーションは JSF テクノロジーを使用して、案内広告データを保管しているデータベースにリンクする動的 Web ページを作成します。JSF コンポーネントを対象のページにドラッグすることによって、CRUD アプリケーション (データベースからの作成、読み取り、更新および削除 (create, read, update, delete) が可能であるため、このように呼ばれます) である JSF Web アプリケーションを作成することができます。サイト・ユーザーは、これらの機能を使用して、データベース内の案内広告を管理できます。

このチュートリアルを使用するためには、オプションでインストール可能ないくつかのコンポーネントが必要な場合があります。適切なオプション・コンポーネントがインストールされていることを確認するには、システム要件リストを参照してください。

このチュートリアルを完了すると、サイト・ユーザーは販売品目を表示できるだけでなく、新規項目を追加したり、項目の詳細 (価格や説明など) を変更したり、特定の項目を検索したりすることができます。このチュートリアル・サイトは単純な設計になっていますが、使用する原理とテクノロジーは、より大規模で複雑な Web サイトでも使用されているものです。

学習目標

このチュートリアルは 2 つのモジュールに分割され、それぞれに学習目標があります。いずれか一方のモジュールを選択することも、あるいは両方のモジュールを選択することもできます。それぞれのモジュールで、Web サイトが適切に作動するように課題を完了してください。

モジュール 1: データ接続を扱う Web ページの作成

このモジュールでは、Web ページで、データベースへの接続をセットアップする方法、およびそのデータベースから情報を表示する方法を学習します。このモジュールでは、以下に挙げることを行います。

- Web ページをデータベースに接続する。
- データベースのレコードを表示、作成、編集、および削除するページを作成する。
- あるページから他のページにデータを送る

モジュール 2: 拡張機能の追加

このモジュールでは、データベースのデータを、より効果的に使用方法を学習します。このモジュールでは、以下に挙げることを行います。

- Web ページでデータベース・レコードをフォーマットする
- Web ページからデータベースへのファイルのアップロードを可能にするコンポーネントを追加する
- 自動的にページ間をナビゲートする
- データベース管理タスクの一部を自動化する

準備が完了したならば、『モジュール 1: データ接続を扱う Web ページの作成』を始めてください。

所要時間

合計: 3 時間 30 分

モジュール 1: 2 時間

モジュール 2: 1 時間 30 分

スキル・レベル

中級

対象読者

Web アプリケーション開発者、Web ユーザー・インターフェース設計者

システム要件

このチュートリアルを完了するには、まずランタイム・サーバーをインストールして構成する必要があります。このチュートリアルは、次のサーバーでテストされました。

- WebSphere® Application Server 6.1

WebSphere Application Servers 以外のサーバー上への SDO のデプロイについては、『ヘルプ・トピック: WAS サーバー以外のサーバーへの SDO のデプロイ (Deploying SDOs on non-WAS servers)』を参照してください。

このチュートリアルでは、製品のインストール中に開始されなかった機能の一部が必要になる場合があります。チュートリアルで説明されているユーザー・インターフェース・オプションが見つからない場合は、適切な機能が使用可能に設定されていることを確認してください (「ウィンドウ」 → 「設定」 → 「機能」)。このチュートリアルを完了するには、次の機能が必要です。

- Java™ デベロッパー
- Web デベロッパー (typical)
- XML デベロッパー
- データベース・デベロッパー
- エンタープライズ Java デベロッパー
- データ

このチュートリアルを使用するためには、アプリケーション・サーバーをインストールして構成しておく必要があります。サーバー・ランタイム環境が使用可能になっていることを確認するには、「ウィンドウ」 → 「設定」をクリックし、「サーバー」を展開して「インストール済みランタイム」をクリックしてください。このペインを使用して、インストール済みのサーバー・ランタイム定義を追加、除去、または編集できます。新規サーバーのサポートをダウンロードしてインストールすることもできます。

前提条件

このチュートリアルを完了するには、以下の項目に精通している必要があります。

- Web サイト、Web ページ、ブラウザー、およびサーバーといった、Web の基本的な設計概念。
- 単純な静的 Web ページの作成方法。
- テーブル、ハイパーリンク、フォーム、およびイメージといった Web ページの要素。
- テーブル、レコード、列、およびフィールドといったデータベースの用語。

以下の知識があると、さらに理解が深まります。

- ワークベンチのパーспекティブおよびビューの使用方法。
- Web ページの HTML コードを編集する方法。

期待される結果

Web アプリケーションが完了すると、ホーム・ページは下図のようになります。サイト・テンプレートによって Web ページに追加される Web サイト・ナビゲーションは、新規広告の作成やすべての広告の表示を行うことができるページにリンクします。以下に表示された表の行アクションを使用して、特定の広告に関連付けられた情報を変更できる更新ページに移動します。

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

関連情報



このチュートリアルの PDF 版の表示

モジュール 1: データ接続を扱う Web ページの作成

このモジュールでは、データベースから情報を表示、編集、削除、作成できる Web ページの作成方法について学習します。JavaServer Faces (JSF) テクノロジーについて学習し、これを使用して複雑なプログラミング操作をシンプルにします。

学習目標

このモジュールでは、Web ページで、データベースへの接続をセットアップする方法、およびそのデータベースから情報を表示する方法を学習します。このモジュールでは、以下に挙げることを行います。

- Web ページをデータベースに接続する。
- データベースのレコードを表示、作成、編集、および削除するページを作成する。
- あるページから他のページにデータを送る

所要時間

このモジュールを完了するには、約 2 時間を要します。チュートリアルを学習しながら動的 Web サービスの他の面を調べるように決めた場合は、終了するのにこれより長くなる場合があります。

演習 1.1: 必須リソースのインポートとターゲット・サーバーの設定

始める前に、このチュートリアル用の必須リソースである Web ページのセットと Derby サンプル・データベースをインポートする必要があります。

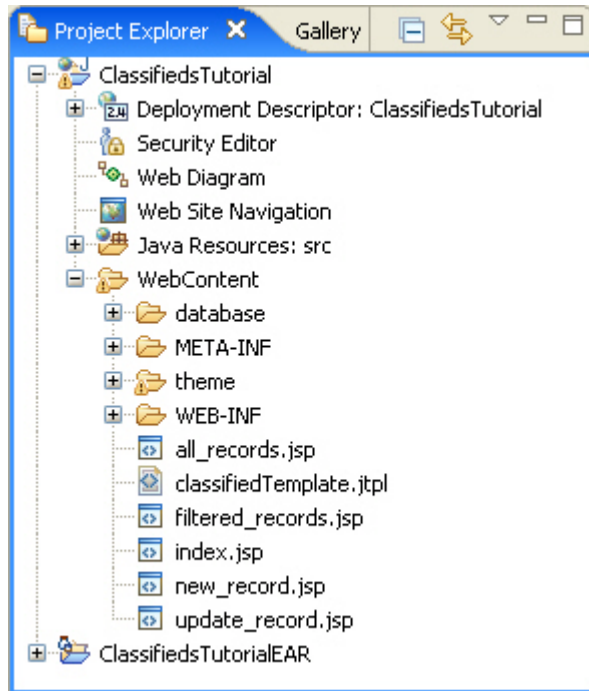
このチュートリアルの主な目的は、ユーザーがデータベース内のデータにアクセスして操作することを可能にする Web アプリケーションの作成方法を学習することです。このチュートリアルでは、Web サイトのルック・アンド・フィールのデザインの詳細については説明しません。従って、Web サイト・デザインはすでに準備済みの状態です。

このチュートリアルを完了するには、既存の JavaServer Faces JSP ファイル・セットおよび Derby サンプル・データベースにアクセスする必要があります。これらの基本的な Web ページとサンプル・データベースが、ZIP ファイルに組み込まれています。この演習では、Web ページおよびデータベースを使用できるようにするために、Zip ファイルのインポート手順を示します。また、動的 Web プロジェクト用のターゲット・サーバーを設定します。

このチュートリアルで使用する Web ページおよびデータベースは、Zip ファイルに入っています。Zip ファイルの内容をインポートする手順は、次のとおりです。

サンプル・プロジェクト・ファイルのインポート

1. プロジェクトをインポートします。Web パースペクティブに切り替えます (「ウィンドウ」 → 「パースペクティブを開く」 → 「Web」)。
2. Web パースペクティブの「プロジェクト・エクスプローラー」ビューで、使用する ClassifiedsTutorial プロジェクトが次のようなイメージになっていることを確認します。



動的 Web プロジェクトのためのターゲット・サーバーの設定

ターゲット・サーバーを Web プロジェクト用に設定し、このチュートリアルで作成するリソースをテストできるようにします。

ターゲット・サーバーを設定する手順は、次のとおりです。

1. Web パースペクティブの「プロジェクト・エクスプローラー」ビューで、「**ClassifiedsTutorial**」を右クリックして「**プロパティ**」を選択します。
2. プロパティ・リストで、「**サーバー**」をクリックします。
3. デフォルト・サーバー・リストで、デフォルトとして使用するサーバーを選択します。「**適用**」をクリックします。
4. プロパティ・リストで、「**ターゲット・ランタイム**」をクリックします。
5. ランタイム・リストで、選択したサーバーに対応するランタイムをクリックします。「**OK**」をクリックします。

使用したいサーバーがターゲット・ランタイム・リストにない場合は、ワークベンチを閉じて該当サーバーをインストールしてください。サーバーのインストールが完了したら、説明に従ってターゲット・サーバーを設定します。

注: サーバー・ランタイムをインストール済みであるのにデフォルト・サーバー・リストにサーバーが一切表示されない場合は、サーバーを構成する必要があると考えられます。サーバーは、次の手順で構成します。

1. 「**all_records.jsp**」ファイルを右クリックして、「**実行**」 → 「**サーバーで実行**」をクリックして選択します。
2. 「**手動で新規サーバーを定義**」を選択します。
3. インストールしたサーバーを選択します。

4. ウィザードの指示に従って、サーバーを構成します。サーバーで初めて実行するときに、エラーを受け取る場合があります。エラーを修正するには、上記のとおりターゲット・サーバーを設定し、「サーバー」ビューでサーバーを再始動してから、ブラウザーに Web ページを再ロードします。

前述のターゲット・サーバー設定のステップを繰り返すと、ここで構成したサーバーがデフォルト・サーバーになっているはずです。インストール済みであっても未構成のサーバーは、デフォルト・ターゲットを選択するためのサーバーのリストに表示されません。

WebSphere Application Servers 以外のサーバー上への SDO のデプロイについては、『ヘルプ・トピック: WAS サーバー以外のサーバーへの SDO のデプロイ (Deploying SDOs on non-WAS servers)』を参照してください。

演習のチェックポイント

これで、ClassifiedsTutorial 動的 Web プロジェクトのインポートと、ターゲット・サーバーの設定が完了しました。

チュートリアル「Web プロジェクト」のファイルを参照することができます。ファイルを開くには、「プロジェクト・エクスプローラー」ビューでそのファイルをダブルクリックします。各ページの関連状況をマップ形式で表示するには、プロジェクト・エクスプローラーで「Web サイト・ナビゲーション」をダブルクリックします。

このサンプルで行う大部分の作業は次のファイルに関連しています。

all_records.jsp

これはサイトのホーム・ページです。データベースにあるすべての案内広告を表示します。

new_record.jsp

このページは新規案内広告を作成します。

update_record.jsp

このページはデータベースにある広告に関する詳細を変更するか、または削除します。

classifiedTemplate.jtpl

これはサイトのページのためのテンプレートです。これには、テーブルや、すべてのページにあるグレーの「広告へようこそ」バナーなどの要素が含まれています。このページには、グレーのバナーの下に 2 つのナビゲーション・タブもあり、ホーム・ページと新規案内広告ページに進むことができます。

これで、『演習 1.2: 関連レコード・リストとデータ・テーブル・コンポーネントを扱う作業』を始める準備が完了しました。

演習 1.2: データベースへの接続と Web ページ上でのデータの表示

この演習では、データベースに接続する方法と Web ページ上でデータ・レコードを表示する方法を学習します。関連レコード・リストを Web ページに追加する方法も学習します。

このチュートリアルの Web サイトは、JavaServer Faces コンポーネントを備えた動的 Web ページを使用してデータベースにアクセスし、情報をページ上に表示します。このチュートリアルでは、関連レコードおよび関連レコード・リストを使用してデータベース内のデータを表し、データ・テーブルや通常の HTML 表の形式でデータをページに表示できるようにします。これらのコンポーネントが Java アクセス Bean を使用します。

データ・アクセス Bean についてさらに学習したい方に:

データ・アクセス Bean は、エンタープライズ Bean を表す Java Bean です。この Bean は、通常、JavaServer Pages (JSP) ファイル、サーブレット、または別のエンタープライズ Bean とインターフェースをとるエンタープライズ Bean を用いるクライアント・プログラムで使用されます。アクセス Bean を使用すれば、エンタープライズ Bean のライフ・サイクルを管理する複雑さから保護されます。言い換えれば、Java Bean のプログラムを作成する場合と同じくらいの容易さで、エンタープライズ Bean のプログラムを作成することができ、エンタープライズ Bean のクライアント・プログラムを簡素化して、全体の開発時間を削減できます。

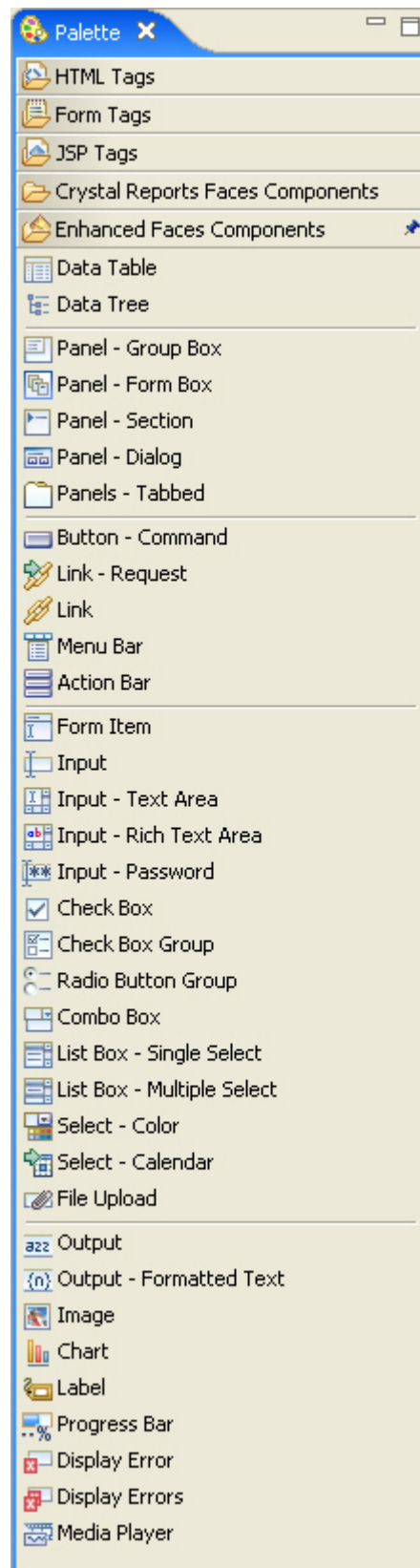
JavaServer Faces および Faces コンポーネントについてさらに学習したい方に:

JavaServer Faces は、サーバーで実行する動的 Web アプリケーションのためのユーザー・インターフェース (UI) のビルドを支援するテクノロジーです。JavaServer Faces フレームワークは、サーバー要求全体にわたって UI 状態を管理し、クライアントが活動化するサーバー・サイド・イベントを開発するための単純なモデルを提供します。JavaServer Faces は Model/View/Controller (MVC) フレームワークを基にしています。これは、JavaServer Faces の場合、コントローラーはサーブレットであり、モデルは JavaBeans™ で表現され、ビューはアプリケーション・コードをほとんど使用しないか、あるいは全く使用しない JavaServer Faces コンポーネントで構成されるということを意味しています。このモデルのゴールは、コンテンツをプレゼンテーションから分離することです。

Faces コンポーネントなどのツールは、ご使用の Web アプリケーションでのこのテクノロジーの使用を支援します。Faces コンポーネントには、ページ内に JavaServer Faces インターフェースを表す JavaServer Page (JSP) カスタム・タグ・ライブラリーが組み込まれています。このウィザードは、Faces コンポーネントの使用を可能にする JSP ファイルの作成を支援します。Faces コンポーネントにより、「パレット」ビューの Faces ドロワーからコンポーネントをドラッグして、作成中の Web ページにドロップすることによって、Web アプリケーションを開発することができます。

例えば、入力 1 行テキスト領域をドラッグして、ページ上のフォームにドロップできます。次に、「実行」ボタンをこの入力 1 行テキスト領域の横にドラッグ・アンド・ドロップできます。最後に、この入力 1 行テキスト領域をユーザーのデータ・ソースに結合することができます。これで、エンド・ユーザーは Web アプリケーションからユーザーのデータ・ソースにデータを入力できるようになります。

さらに、Faces コンポーネントを使用してビルドされたアプリケーションには、その基礎であるプログラム・ロジックとは独立して、ユーザー・インターフェースをレンダリングできる利点があります。このことは、ユーザーのアプリケーションが、ブラウザーまたはハンドヘルド・デバイスなどのさまざまなプラットフォーム上で実行し、データにアクセスできることを意味しています。



関連レコード・リストの追加

この演習では、データベースにあるすべての案内広告を表示するための関連レコード・リストを作成します。次に、データベースに接続して、関連レコード・リストの中で必要とする情報を保持しているテーブルを選択します。最後に、この関連レコード・リストを、データ・テーブルのページに表示します。

- 関連レコードは、データベースの 1 つのレコードだけに接続します。このケースでは、1 つの関連レコードは、サンプル・データベースの単一案内広告を表します。関連レコードを使用すれば、新規レコードの作成、既存レコードの編集、または古いレコードの削除を行えます。
- 関連レコード・リストは、データベースの複数のレコードに接続します。このケースでは、関連レコード・リストはサンプル・データベースの 2 つからすべてまでの任意の数の案内広告を表示します。
- データ・テーブルは、関連レコード・リストからデータをページに表示します。データ・テーブルは、単にレコード・リストの場所を指定するだけであり、HTML 表と同じようにデータを行と列にフォーマット設定するわけではありません。

データ・テーブルについてさらに学習したい方に:

データ・テーブルは、関連レコード・リストのようなデータ・オブジェクトを保持する Faces JSP ページのコンポーネントです。データ・テーブルには HTML 表のように行と列があるように見えますが、これは HTML 表のように機能しません。テーブル内と同様に入出力コントロールをフォーマットしたい場合は、パレットの拡張 Faces コンポーネント・ドロワーから、「パネル - グループ・ボックス」を使用します。この方法は『演習 2.2: データ・テーブルのフォーマット』に説明してあります。

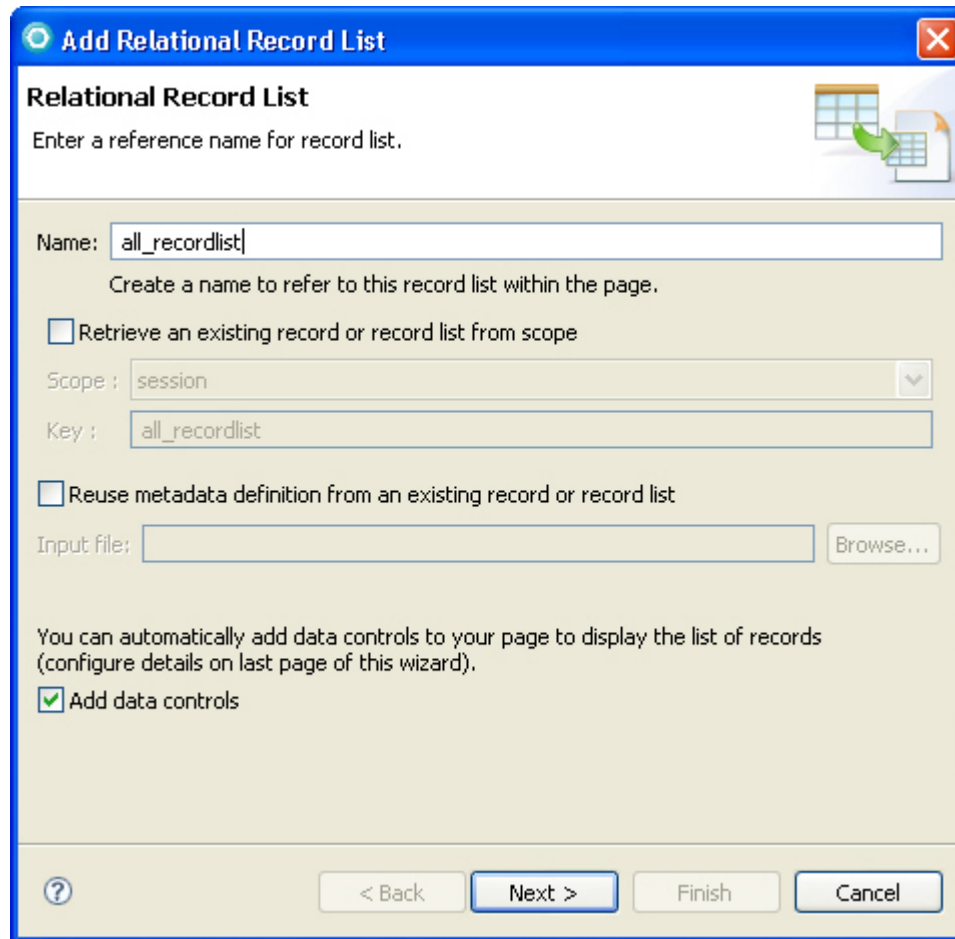
データ・テーブルは、Faces コンポーネントであって、HTML コンポーネントではないので、「Page Designer」ビューではなく、「プロパティ」ビューでコントロールされます。「プロパティ」ビューを使用して、以下のようにデータ・テーブルをさまざまな方法でカスタマイズすることができます。

- 列順を追加、除去、および変更する
- テキストおよび背景をフォーマットする
- ヘッダー、フッター、および余白を追加する
- 結果を表示するページングを追加する

新規関連レコード・リストを作成するには、以下のとおりに実行します。

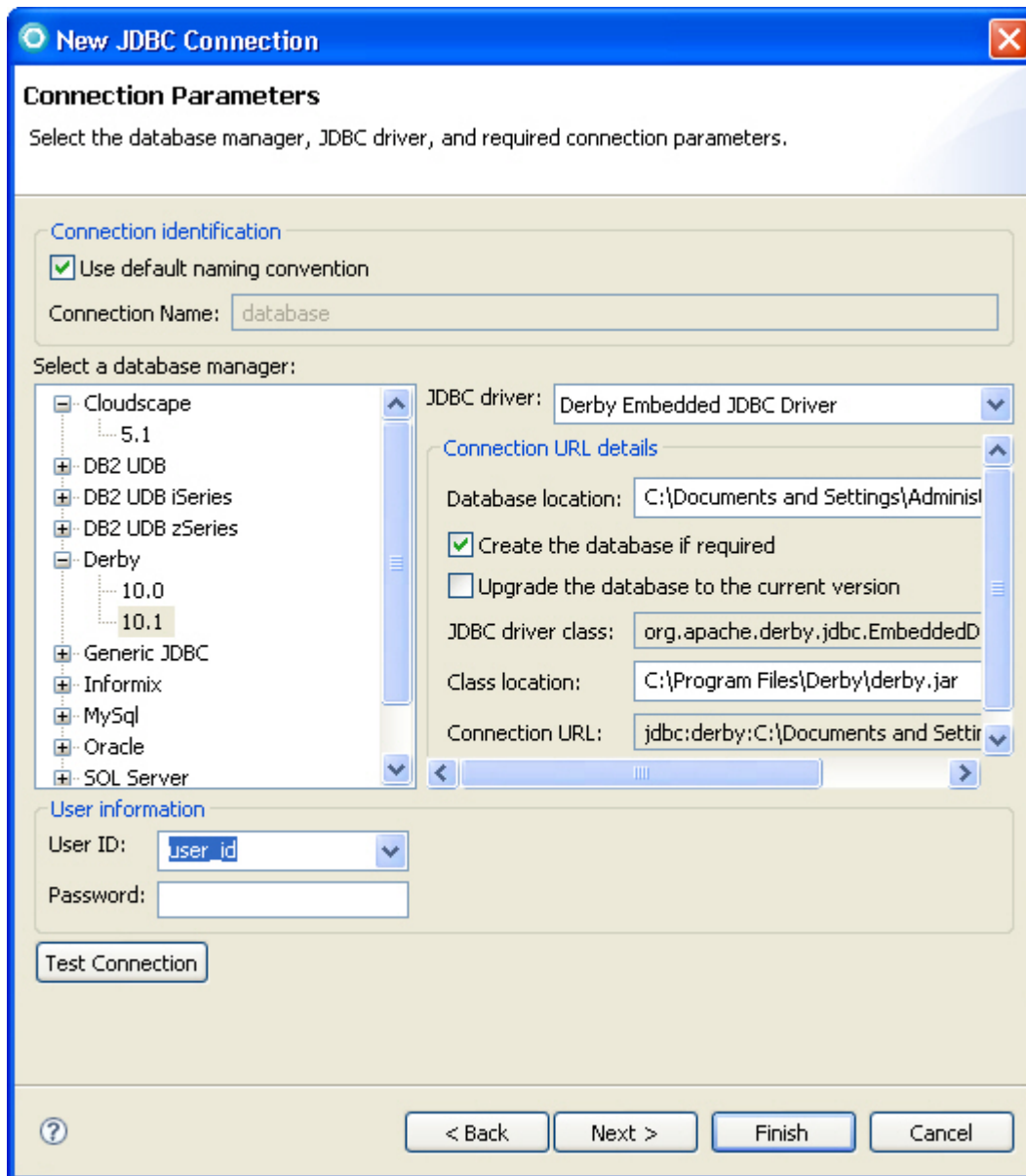
1. Web パースペクティブの「プロジェクト・エクスプローラー」ビューで、「**ClassifiedsTutorial**」 → 「**WebContent**」を展開する。
2. 「WebContent」フォルダーで、「**all_records.jsp**」をダブルクリックする。エディターで、**all_records.jsp** ファイルが開きます。
3. デフォルトのテキスト「bodyarea のデフォルトのコンテンツ (Default content of bodyarea)」を削除する。
4. 「パレット」ビューで**データ・ドロワー**を展開する。
5. パレットから「**関連レコード・リスト**」コンポーネントをブランクのコンテンツ領域にドラッグする。「**関連レコード・リストの追加**」ウィンドウが開きます。
6. 「名前」フィールドに「**all_recordlist**」と入力する。 関連レコード・リストおよび関連レコード名は、変数名に関する Java 標準命名規則に準拠している必要があります (例えば、スペースを含むことはできません)。
7. 「**データ・コントロールの追加**」が選択されていることを確認する。「**データ・コントロールの追加**」が選択されていれば、ウィザードによりデータ・テーブルが作成され、レコード・リストがページに表示されます。選択されていない場合、ウィザードはレコード・リストを作成するだけで、ページ上

にそのデータは表示しません。ここでは、ウィザードにデフォルトのデータ・テーブルを作成させ、ユーザーは後からそれをカスタマイズします。「関連レコード・リストの追加」ウィンドウは、次のようになるはずです。

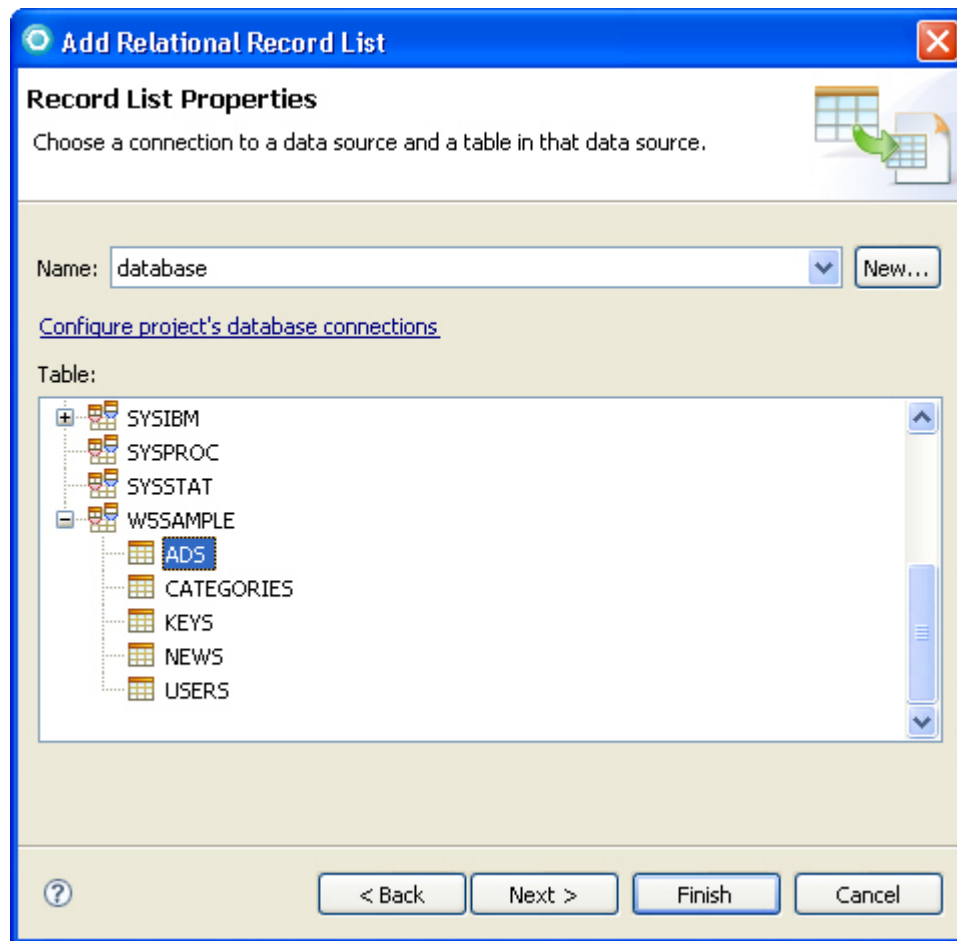


「次へ」をクリックする。

8. ウィザードの次のページにある「接続名」フィールドで、「新規」をクリックして新規データベース接続を作成する。「新規 JDBC 接続」ダイアログが開きます。「新規接続の作成」ラジオ・ボタンを選択し、「次へ」をクリックする。
9. 「データベース・マネージャーの選択」リストで、「Derby」を展開し、次に「10.1」を選択する。
10. 「データベース・ロケーション」フィールドで、「参照」をクリックし、「<workspace_location>%ClassifiedsTutorial%WebContent%database」を選択する。ここで、「<workspace_location>」は現行ワークスペースのディレクトリーです。「OK」をクリックして、「フォルダーの参照」ウィンドウを閉じます。
11. 「クラス・ロケーション」フィールドで、JDBC ドライバー・クラスを実装する JAR (複数可) のロケーションを入力するか、「参照」をクリックしてファイル・システムからロケーションを選択する。JAR ファイルのロケーションを選択したら、「開く」をクリックします。
12. データベースにアクセスするには、ユーザー ID の入力が必要となる場合があります。パスワードは必要ありません。「新規データベース接続」ウィザードは、次のようになるはずです。



13. 「新規 JDBC 接続」ウィザードで、「終了」をクリックする。「関連レコード・リストの追加」ウィザードに戻ります。
14. 「プロジェクトのデータベース接続の構成」をクリックする。「JDBC 接続プロパティ (JDBC Connections properties)」ウィンドウが開きます。
15. 「ランタイム接続の詳細」セクションの隣にある「編集」をクリックする。ランタイム接続のタイプとして「ドライバ・マネージャー接続の使用」を選択する。「終了」をクリックし、「OK」をクリックすると、「関連レコード・リストの追加」ダイアログに戻ります。Derby データベースへの接続を作成したので、表示するテーブルまたはレコード・リストを選択する必要があります。「関連レコード・リストの追加」ウィザードには、データベースのテーブルが表示されます。
16. 「W5SAMPLE」を展開し、「ADS」を選択する。「次へ」をクリックする。このウィザードの残りのページで、ユーザーのレコード・リストから列を除外して、拡張オプション (例えば、主キーの定義、他のテーブルへの関係の追加、フィルターおよびソート条件の指定) を実行することができます。これらのページについては、後続の課題でさらに学習します。

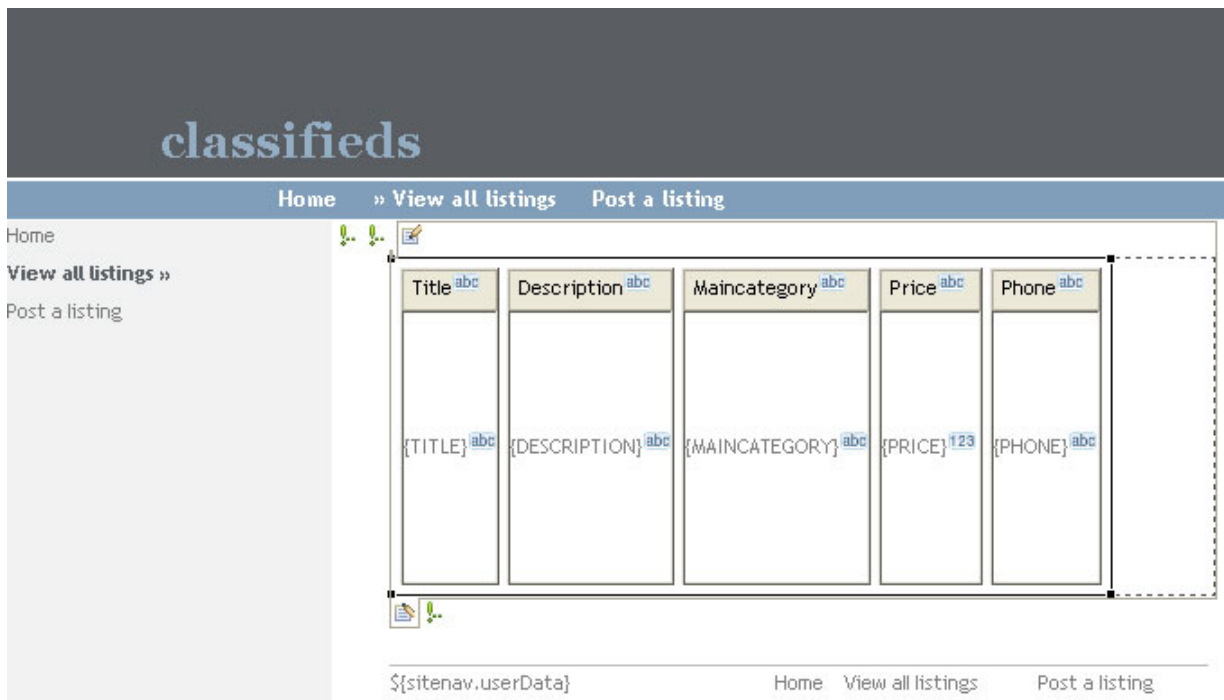


17. ダイアログの「関連レコードの追加」ページで、関連レコード・リストに組み込む列を指定する。「次へ」をクリックします。
18. このチュートリアルで表示すべき項目は、各案内広告のタイトル、説明、カテゴリー、価格、および電話番号のみです。デフォルトでは、すべての列が選択されています。以下を除くすべての列をクリックしてクリアし、表示を抑止します。
 - **MAINCATEGORY: String**
 - **TITLE: String**
 - **DESCRIPTION: String**
 - **PRICE: BigDecimal**
 - **PHONE: String**
19. Web ページでのデータ・テーブル表示の編成を洗練させることができる。列を選択し、上矢印および下矢印をクリックして望ましい位置に再配列することができます。

列を次の順序に配置します。

- **TITLE**
- **DESCRIPTION**
- **MAINCATEGORY**
- **PRICE**
- **PHONE**

「終了」をクリックします。 Web ページは、次のように表示されるはずです。



20. このページを保管する。

次の演習では、このページが実際の Web サーバーでどのように表示されるかを確認します。データ・テーブルと JavaServer Faces コンポーネントをフォーマットするには多くのオプションがあります。これらのオプションの一部は、次のモジュールで取り上げています。ページ上の種々の JavaServer Faces コンポーネント (例えば、データ・テーブル、および個々の出力コンポーネント) について、独自に「プロパティ」ビューを調べてみることもできます。

演習のチェックポイント

演習 1.2 が完了しました。この演習では、データベースに接続する方法と、Web ページ上で関連レコード・リストを使用してデータ・レコードを表示する方法を学習しました。

演習 1.3: Web サイトのテスト

Web アプリケーションを公開する準備が完了したら、それをホストして、ユーザーがインターネット経由で Web サイトにアクセスできるようにするサーバーが必要になります。一方、Web サイトのテスト時には、テスト環境を使用できます。

この演習では、テスト環境で JSP を実行して、実際に公開したときにページがどのように表示されるかをシミュレートします。サーバーの開始および停止方法も学習します。Web サイト開発中は、随時、ページを Page Designer で開き、「プレビュー」タブを使用して、ユーザーの設計がブラウザーでどのように見えるかを表示することができます。ただし、「プレビュー」ビューでは、ユーザーのページの動的局面 (例えば、データベース接続) を、サーバーでの実行を表示するように示すことはできません。

ヒント: このサーバーが開始されたならば、このサイトで作業を継続できるようになるには、まずサーバーを停止する必要があります。

このチュートリアルを使用するためには、アプリケーション・サーバーをインストールして構成しておく必要があります。サーバー・ランタイム環境が使用可能になっていることを確認するには、「ウィンドウ」 →

「設定」をクリックし、「サーバー」を展開して「インストール済みランタイム」をクリックしてください。このペインを使用して、インストール済みのサーバー・ランタイム定義を追加、除去、または編集できます。新規サーバーのサポートをダウンロードしてインストールすることもできます。

Web サイトをサーバーでテストするには、次のようにします。

1. 「プロジェクト・エクスプローラー」ビューで「**all_records.jsp**」ファイルを右クリックして、「**実行**」→「**サーバーで実行**」を選択します。「サーバーで実行」ウィンドウが開きます。
2. 演習 1.1 でターゲット・サーバーとして選択したのと同じサーバーを使用する必要があります。このステップで別のサーバーを使用すると、エラーとなります。
 - 事前に定義したサーバー構成で Web サイトをテストする手順は、次のとおりです。
 - a. 「**既存のサーバーを選択**」を選択する。
 - b. 使用するサーバーを使用可能なリストから選択する。
 - 新しいサーバー構成で Web サイトをテストする手順は、次のとおりです。
 - a. 「**手動で新規サーバーを定義**」を選択する。
 - b. 使用するサーバー・タイプを使用可能なリストから選択する。
3. 「**終了**」をクリックする。

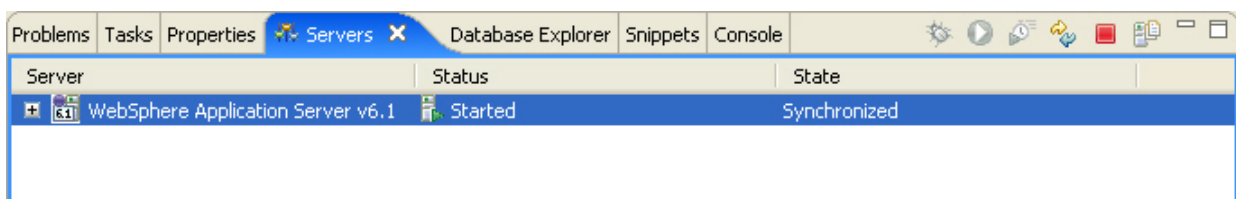
サーバー・ツールは新規テストを作成して、開始し、さらにワークベンチ内部 Web ブラウザーでページを開きます。これにはしばらく時間がかかります。サーバー・ツールがサーバーを開始するとき、「コンソール」ビューで、メッセージを監視できます。

このプロセスが完了したら、このサイトが実際に Web サーバーに公開されるときにどのように見え、どのように作動するかについてプレビューできます。リンクをクリックし、データをフォームに入力して、「プレビュー」ビューには表示されないことがある動的に生成されたコンテンツをサイト上に表示させます。さらに、別の Web ブラウザーでユーザーのページを見たい場合は、ワークベンチの Web ブラウザーから URL を、選択したブラウザーにコピーします。

このチュートリアルを続行できるようにするには、まずサーバーを停止する必要があります。サーバーを実行したままにしておくと、このチュートリアルを続行するときにエラー・メッセージが表示されます。これらのエラーは、サーバーがサンプル・データベースに接続されているために現れるもので、Web サイトに表示された情報を変更するためにデータベースに接続することができなくなります。

サーバーを停止するには、次のようにします。

1. 「サーバー」ビューを開く。このビューは、通常ワークベンチの下部中央に配置されています。「サーバー」ビューを見つけることができない場合は、メインメニューから「**ウィンドウ**」→「**ビューの表示**」→「**サーバー**」をクリックします。
2. 停止するサーバーを選択する。サーバー状況が「**始動済み**」としてリストされていることに注意してください。



3. 「サーバー」ビューの右上部にある「サーバーの停止」ボタン をクリックする。サーバー「状況」は「停止中」に変更されます。サーバーのシャットダウン中は、「コンソール」ビューでメッセージが表示されます。「サーバー」ビューでサーバー「状況」が「停止済み」に変更されると、サーバーは停止し、Web サイトでの作業を続行できます。

ヒント: ユーザーの Web サイトはいつでもサーバーでテストできますが、完了したときにサーバーを停止することを忘れないでください。

演習のチェックポイント

演習 1.3 が完了しました。この演習では、テスト環境で JSP をテストして、実際に公開したときにページがどのように表示されるかをシミュレートしました。サーバーの開始および停止方法も学習しました。

演習 1.4: 表示する新規レコードの作成とデータベースの更新

この演習では、ユーザーが新規案内広告を作成してそれをデータベースに掲載することができるページを作成します。

最初に、新規レコードをデータベースに表すための関連レコードを作成し、次に 1 セットの入力フィールドとして、ユーザーのページにレコードを視覚化したものを作成します。

新規関連レコードを作成するには、以下のとおりに実行します。

1. 「プロジェクト・エクスプローラー」ビューの「**new_record.jsp**」ファイルをダブルクリックして開く。
2. デフォルトのテキスト「bodyarea のデフォルトのコンテンツ (Default content of bodyarea)」を削除する。
3. 「パレット」ビューで、**データ・ドロワー**をクリックしてそれを展開する。
4. パレットから「**関連レコード**」コンポーネントをブランクのコンテンツ領域にドラッグする。「関連レコードの追加」ウィンドウが開きます。

注: 使用するデータベースへの接続が確立できなかったことを示す警告メッセージが表示された場合は、Web サイトのテスト後、サーバーを実行したままにしておいたのです。この警告メッセージが表示された場合は、それぞれのダイアログの「**キャンセル**」をクリックして、『演習 1.3』の説明に従ってサーバーを停止してください。

5. 「名前」フィールドに「create_record」と入力する。
6. 「**コントロールの作成対象**」の下「**新規レコードの作成**」をクリックする。
7. 「**レコードを Web ページに表示するための入出力コントロールの追加**」が選択されていることを確認する。「関連レコードの追加」ウィンドウは、次のようになるはずです。

8. 「次へ」をクリックする。
9. 「テーブル」ボックスで「W5SAMPLE」を展開し、「ADS」を選択する。「次へ」をクリックします。
10. 再度「次へ」をクリックする。「データ・コントロールの構成」ページが開きます。

ヒント: 「データ・コントロールの構成」ページは、ユーザーの関連レコードの視覚化のカスタマイズを支援します。例えば、ユーザーの入力フォームにある列、フィールド・ラベル、および「実行」ボタンを変更できます。以下の手順を終了すると、全機能の入力フォームがページに作成されます。

11. 「表示するフィールド」セクションで、ユーザーの入力フォームに使用したいフィールド名を除いて、その他すべてのフィールド名の横にあるチェック・ボックスをクリアする。
 - DESCRIPTION
 - ID
 - MAINCATEGORY
 - PHONE
 - PRICE
 - TITLE

12. 上矢印または下矢印のボタンをクリックすることにより、次の順序でフィールドを再配列する。
 - ID
 - TITLE
 - DESCRIPTION
 - MAINCATEGORY
 - PRICE
 - PHONE
13. 自分の望みに合わせてラベルを名前変更する。例えば、「Maincategory:」ラベルを「Category:」に短縮します。入力フィールド用に生成されたラベルを名前変更するには、「ラベル」列から該当のラベルをクリックします。マウス・アイコンがカーソルに変わり、新規テキストを入力できるようになります。
14. 「**MAINCATEGORY**」を選択して、「**オプション**」をクリックする。「オプション」ウィンドウが開きます。
15. 「**実行ボタン**」オプションが選択されていることを確認する。
16. 「ラベル」フィールドに「新規リストの掲載 (Post New Listing)」と入力する。次に「**OK**」をクリックします。「関連レコードの追加」ウィンドウは、次のようになるはずです。

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.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

17. 「終了」をクリックして、入力フォームを生成する。このフォームは、次のようになるはずです。

The screenshot shows a web application titled 'classifieds'. The main content area is a form for posting a new listing. The form has 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 section labeled '[Error Messages]' with a small red icon and a blue arrow pointing down. At the bottom of the form is a button labeled 'Post New Listing'.

The page layout includes a dark header with the 'classifieds' logo, a blue navigation bar with links to 'Home', 'View all listings', and '» Post a listing'. A sidebar on the left contains links to 'Home', 'View all listings', and 'Post a listing »'. The footer shows a user data field labeled '\${sitenav.userData}' and navigation links to 'Home', 'View all listings', and 'Post a listing'.

注: このフォームには「エラー・メッセージ」フィールドがあります。これは、プロジェクトにエラーがあることを意味するものではありません。このフィールドは、ユーザーがフォームを実行したときにエラーがあった場合、そのエラーが表示される場所をマークします。

18. このページを保管する。

入力フォームが実行されると、ページは新規レコードをデータベースに自動的に追加します。次の演習では、「新規リストの掲載 (Post New Listing)」ボタンを「all_records.jsp」ページに戻るようにプログラミングして、データベース内の新規レコードを即時に表示できるようにします。

演習のチェックポイント

演習 1.4 が完了しました。この演習では、ユーザーが新規案内広告を作成してそれをデータベースに掲載することができるページを作成しました。

演習 1.5: 「実行」ボタンのプログラミング

入力フォームが実行されると、ページは新規レコードをデータベースに自動的に追加します。この演習では、「新規リストの掲載 (Post New Listing)」実行ボタンを「all_records.jsp」ページに戻るようにプログラミングして、データベース内の新規レコードを即時に表示できるようにします。

実行ボタンは、次の手順でプログラミングします。

1. new_record.jsp ファイルで、Web ページのフォームで作成した「新規リストの掲載 (Post New Listing)」ボタンを右クリックし、「プロパティ」を選択する。
2. 「規則の追加」をクリックする。「ナビゲーション規則の追加」ウィンドウが開きます。
3. 「ページ」ドロップダウンから「all_records.jsp」を選択して「OK」をクリックする。
4. このページを保管する。レコードをデータベースに追加して all_records.jsp ページに戻ることを検証する場合は、テスト・サーバーで new_record.jsp を実行します。

重複キーの防止

重要:

ID 列は ADS テーブルの主キーであるので、このテーブルに既存の ID 値を持つレコードはここに追加できません。各新規レコードのために新規の未使用キーを自動的に作成するために自動キー生成を使用する方法については、モジュール 2 で学習します。

その方法を学習するまでは、新規レコードの追加には、未使用 ID 番号をこのページに入力する必要があります。このデータベースが持っているレコードは 1 から 24 までの ID 番号を使用しているので、主キーとしては 24 を超える番号を入力します。複数のレコードを入力する場合は、キーが重複していないことを確認してください。

入力の関連レコードへのバインディング (オプション)

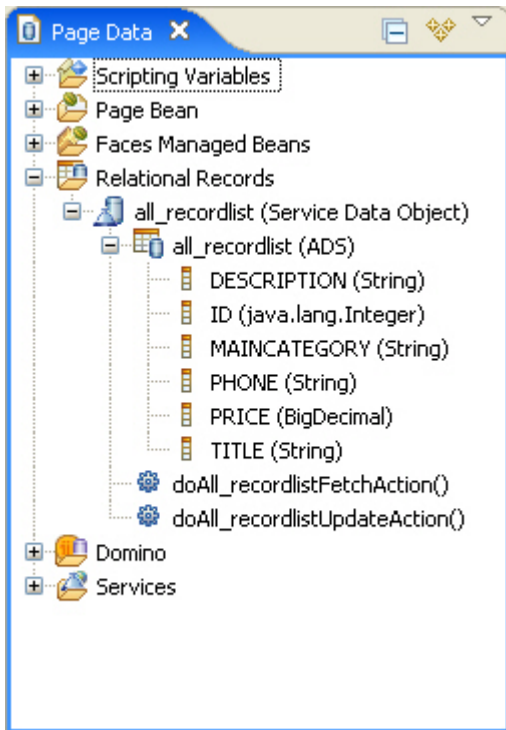
ページの入力フォームとは、作成した関連レコードにバインドされた 1 組の JavaServer Faces 入力フィールドのことです。使用するデータベースの ADS テーブルに新規レコードを表示するために、関連レコードを作成したことを思い出してください。バインディング とは、JavaServer Faces 入力コンポーネントを関連レコードの列にリンクできる方法のことです。

関連レコードを作成すると、ウィザードはその列のすべてを、ページにある入力フィールドに自動的にバインドします。手動で変更を行いたい場合は、それぞれの列をそれぞれ別の入力フィールドにバインドすることができます。ユーザーの関連レコードの列を入力フィールドにバインドするには、「ページ・データ」ビューから列をこのフィールドにドラッグします。このプロセスは、ユーザーのフォームにある「説明」入力フィールドを削除してから再作成することによって実験してみることができます。

「ページ・データ」ビューについてさらに学習したい方に:

「ページ・データ」ビューは、普通、ワークベンチの左下隅にあります。見つけれない場合は、メニュー・バーから「ウィンドウ」→「ビューの表示」→「ページ・データ」と順にクリックします。一般に、「ページ・データ」ビューは動的 Web ページを作成するときのみ使用します。

「ページ・データ」ビューは、関連レコードまたは関連レコード・リストといったデータ・オブジェクトの形式で、データベースのようなデータ・ソースへの関連付けを保管しています。これらのデータ・オブジェクトはデータ・ソース自体も、あるいは、ページ上のいずれのコンポーネントも表していません。データ・オブジェクトは、それらを表すのではなく、プロジェクトとデータ・ソースの間の接続を表します。この接続が作成されると、「ページ・データ」ビューからデータ・オブジェクトを Web ページにドラッグして、この接続を完成させ、データ・ソースの情報を Web ページに表示できます。例えば、次に示すのは、このチュートリアルで作成された関連レコード・リストを示す「ページ・データ」ビューの図です。



この図には、「all_recordlist (ADS)」関連レコード・リストと、それに続いて、この関連レコード・リストにあるすべての列を見ることができます。ひとたびこのデータ・オブジェクトが定義されたならば、データベースに再接続せずに「ページ・データ」ビューで何回でも、これらの列を Web ページの入力コンポーネントにドラッグすることができます。

例えば、DESCRIPTION 列の値を Web ページに表示したい場合、最初にテキスト出力フィールドをページにドラッグし、次に「ページ・データ」ビューの「DESCRIPTION」列をそのテキスト出力フィールドにドラッグします。このことを「列を出力フィールドにバインドする」と表現します。

あるいは、ユーザーにデータベースの DESCRIPTION 列のための新規の値を入力させたい場合は、テキスト入力フィールドをこのページにドラッグすることもできます。次に、「DESCRIPTION」列をそのテキスト入力フィールドにドラッグします。

以下の手順は入力フィールドの概念およびバインディングのプロセスを説明するために準備されたものですが、これらの手順をウォークスルーすることは、このチュートリアルではオプションです。ウォークスルーしない場合は、次の演習に進んでください。

1. new_record.jsp ファイルで、「説明」入力フィールドをクリックする。「削除」を押す。
2. 「パレット」ビューで、拡張 **Faces** コンポーネント・ドロワーをクリックして展開する。
3. パレットから「入力」コンポーネントを、削除した「説明」入力フィールドが入っていたセルにドラッグする。これで、入力フィールドがこのセルに入りますが、この入力フィールドはいずれの列にもバインドされていないので、その中には {ID} または {TITLE} のようなテキストは入っていません。
4. 「ページ・データ」ビューで、「関連レコード」→「create_record (サービス・データ・オブジェクト)」→「create_record (ADS)」をクリックして展開し、作成した「入力」コンポーネントに「DESCRIPTION」列をドラッグする。「入力」コンポーネント内のテキストが変更され、「DESCRIPTION」列にバインドされていることがわかります (下図参照)。

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

5. このファイルを保管して、次に必要であれば、ページをテスト・サーバーで実行する。

演習のチェックポイント

演習 1.5 が完了しました。この演習では、コマンド・ボタンをプログラミングし、パレットから Web ページに JSF ウィジェットをドラッグ・アンド・ドロップすることで、データを入力フィールドにバインドすることを学習しました。

演習 1.6: 更新ページの作成

ここまでで、Web サイトのために、リストを表示し、作成するページが作成できています。この演習では、ユーザーがリストを更新したり、削除したりすることを可能にするページを作成します。更新ページは、ほとんど寸分たがわず作成ページと同じように見えます。ただし、更新ページでは、ユーザーが変更できるように、入力フィールドに既存のレコードからのデータが表示される点が異なります。

最初に、データベースからの既存のレコードを表示する関連レコードを作成します。次に、この関連レコードのための JavaServer Faces 更新フォームを作成して一部を変更すると、このページはテストの準備が整います。

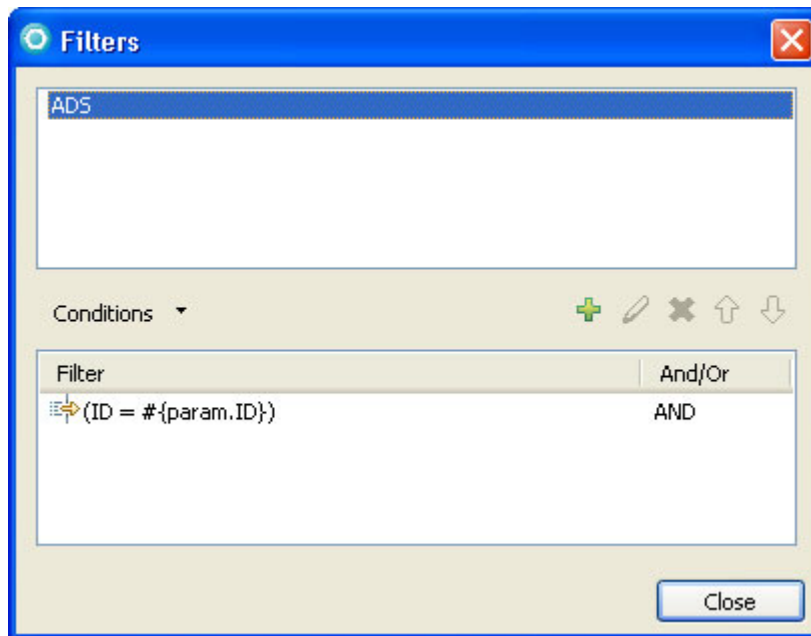
更新関連レコードの作成

1. 「プロジェクト・エクスプローラー」ビューで、「update_record.jsp」ファイルをダブルクリックして開く。
2. デフォルトのテキスト「bodyarea のデフォルトのコンテンツ (Default content for bodyarea)」を削除する。
3. 「パレット」ビューで、**データ・ドロワー**をクリックしてそれを展開する。
4. パレットから「**関連レコード**」コンポーネントをブランクのコンテンツ領域にドラッグする。「関連レコードの追加」ウィンドウが開きます。
5. 「名前」フィールドに「update_record」と入力する。
6. 「**コントロールの作成対象**」の下「**既存レコードの更新**」をクリックする。
7. 「**関連レコードを Web ページに表示するための入出力コントロールの追加**」が選択されていることを確認する。
8. 「**次へ**」をクリックする。
9. 「**テーブル**」ボックスで「W5SAMPLE」を展開し、「ADS」を選択する。
10. 「**次へ**」をクリックする。「**関連レコードの追加**」ページが開きます。

結果のフィルター操作

関連レコードはデータベースから 1 レコードのみ表示できます。したがって、ユーザーの編集用に 1 レコードのみが表示されるようにデータベース・テーブルをフィルター操作する必要があります。(前の演習ではデータベースのフィルター操作をする必要はありませんでした。それは、新規レコードは 1 つしか作成しなかったため、データベースにはフィルター操作すべき結果がなかったからです。) データベースのそれぞれのレコードには固有 ID 番号が付けられているので、結果をフィルター操作して、特定の ID 番号を持つレコードを得ることができます。

1. 「関連レコードの追加」ページには、タスク・リンクのリストがある。「タスク」の下に「フィルター結果」をクリックする。「フィルター」ウィンドウが開くので、デフォルトのフィルター条件「ID = #{param.ID}」を「フィルター」列に挿入する。「フィルター」ウィンドウは、次のようになります。



このコードは、データベースのレコードをフィルター操作します。したがって、指定済み ID 番号を持つレコードのみが関連レコードに表示されます。この条件に関する詳細は、この演習の後半にある『行アクションの追加』セクションで学習します。「閉じる」をクリックする。

2. 「次へ」をクリックする。「データ・コントロールの構成」ページが開きます。
3. 「表示するフィールド」セクションで、対象の入力フォームに表示したいフィールドを除くすべてのフィールド名の横にあるチェック・ボックスをクリアする。
 - ID
 - TITLE
 - DESCRIPTION
 - MAINCATEGORY
 - PRICE
 - PHONE
4. 「上へ」または「下へ」をクリックして、フィールド名を以下のように上から下へと再配列する。
 - ID
 - TITLE
 - DESCRIPTION

- MAINCATEGORY
 - PRICE
 - PHONE
5. ID フィールドについて、「コントロール・タイプ」列のドロップダウン・リストから「出力フィールド」を選択する。レコード ID 番号は、ユーザーが表示できるようにしますが、ユーザーが更新できるようにはしたくありません。ID フィールドを出力フィールドにすると、重複 ID の問題を回避するのに役立ちます。
 6. 「オプション」をクリックする。「オプション」ウィンドウが開きます。「実行ボタン」オプションが選択されていることを確認します。「ラベル」フィールドに「更新」と入力します。「OK」をクリックします。「関連レコードの追加」ウィンドウは、次のようになるはずです。

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.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

7. 「終了」をクリックして、下図のようにページに更新フォームを生成する。

The screenshot shows a web application titled "classifieds". It has a navigation bar with links: Home, View all listings, and Post a listing. On the left, there is a sidebar with the same links. The main content area displays a form for editing a record. The form has the following fields:

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

Below the form, there is a section for error messages, followed by "Update" and "Delete" buttons. At the bottom of the page, there is a footer with the text "\${sitenav.userData}" and the same navigation links: Home, View all listings, and Post a listing.

「更新」および「削除」ボタンのプログラミング

ここでもう一度、ユーザーが `all_records.jsp` ページを参照し、その他のレコードとともに変更したレコードを表示するためのコードを追加します。

1. Web ページで作成した「更新」ボタンを右クリックして、「プロパティ」を選択する。
2. 「規則の追加」をクリックする。「ナビゲーション規則の追加」ウィンドウが開きます。
3. 「ページ」ドロップダウンから「`all_records.jsp`」を選択する。「この規則の使用条件」ボックスで、「このアクションのみ」をクリックする。フィールドに `#{pc_Update_record.doUpdate_recordUpdateAction}` が含まれていることを確認してください。「OK」をクリックします。
4. Web ページ上の「削除」ボタンのために、ステップ 1 から 3 を繰り返して、同じようにコードを編集する。「この規則の使用条件」フィールドに `#{pc_Update_record.doUpdate_recordDeleteAction}` が含まれていることを確認してください。
5. このページを保管する。レコードをデータベースに追加して `all_records.jsp` ページに戻ることを検証する場合は、テスト・サーバーで `new_record.jsp` を実行します。

行アクションの追加

ここで、更新するデータベース・レコードをユーザーが選択できるように、`all_records.jsp` に表への行アクションを追加します。「`#{param.ID}`」は、更新ページが更新するレコードの ID 番号を表します。ユーザーが行をクリックすると、そのレコード ID 番号は「`#{param.ID}`」パラメーターとして `update_record.jsp` ページに送られます。そこで、`update_record.jsp` ページに挿入されたばかりのフィルター済み関連レコードは、更新用のそのレコードのみを表示します。

1. 「プロジェクト・エクスプローラー」ビューで、「`all_records.jsp`」ファイルをダブルクリックして、エディターを開く。データ・テーブル内の任意の場所をクリックする。「プロパティ」ビューを開く。
2. 「プロパティ」ビューで、HTML タグ・リストから「`hx:dataTableEx`」での「行アクション」タブをクリックする。

3. 「行がクリックされたときに実行されるアクションを追加 (Add an action that's performed when a row is clicked)」をクリックする。「行アクションの構成」ダイアログが開きます。「行をクリックすると、サーバーに要求が送信され、行情報もパラメーターとして送信される (Clicking the row sends a request to the server with row information sent as parameters)」を選択し、次に「OK」をクリックします。列がデータ・テーブルに追加されます。
4. 「プロパティ」ビューで、「hx:requestRowAction」タブをクリックする。
5. 「規則の追加」をクリックする。「ナビゲーション規則の追加」ダイアログが開きます。
6. 「ページ」ドロップダウンから「update_record.jsp」を選択する。「この規則の使用条件」ボックスで、「このアクションのみ (This action only)」を選択します。「このアクションのみ」フィールドには、行アクションを追加した際に作成される `#{pc_All_records.doRowAction1Action}` が自動的に取り込まれます。「OK」をクリックします。このフィールドに上記のメソッドが自動的に取り込まれない場合は、フィールドにメソッドを入力し、ClassifiedsTutorial/Java Resources「src/pagecode/All_records.java」に次を追加してください。

```
public String doRowAction1Action() {  
    return "";  
}
```
7. 「プロパティ」ビューで、「hx:requestRowAction」での「パラメーター」タブをクリックする。「パラメーターの追加」をクリックし、次に「名前」フィールドに「ID」と入力します。この ID パラメーターを ID 列にバインドする必要があります。この行アクションをデータ・リストの ID 列にバインドすることは、行がクリックされるときに、要求パラメーターがリストのレコード ID になることを意味します。
 - a. 「値」をクリックし、次に「ページ・データ・オブジェクトの選択」  ボタンをクリックする。
 - b. 「データ・オブジェクト」タブで、「all_recordlist (サービス・データ・オブジェクト)」 → 「all_recordlist (ADS)」を展開し、次に「ID (java.lang.Integer)」を選択する。
 - c. 「OK」をクリックします。

これで、ユーザーは、行をクリックすると、Web サイトで案内広告に関する情報を更新できるようになります。このファイルを保管し、必要であればページをテストする。最初に all_records.jsp を開いておくことを忘れないでください。この JSP が update_record.jsp にリンクするページだからです。

演習のチェックポイント

演習 1.6 が完了しました。この演習では、データベースのレコードを更新する Web ページの作成方法を学習しました。

モジュール 1: 要約

『モジュール 1: データ接続を扱う Web ページの作成』が完了しました。

学習した内容

次のことを学習しました。

- データベースの情報を Web ページに表示する。
- 関連レコード、関連レコード・リスト、およびデータ・テーブルを処理する。
- データベース・レコードを Web ページから表示、編集、作成、削除する。
- データ・テーブル・コンポーネントの行アクションを使用してレコード、またはレコード・セットにリンクする。

このモジュールでは、JavaServer Faces テクノロジーおよびツールについて紹介しました。サイトを魅力的で、使いやすく、効率的なものにするために、さらに多くの手段があります。

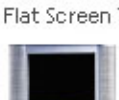
次のモジュールの『モジュール 2: 拡張機能の追加』では、この単純な案内広告サイトを、よりビジュアル的に楽しく、さらに複雑なアプリケーションに変更する方法について学習します。ご使用のサイトを次のような表示にしたい場合は、次のモジュールに進んでください。

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

モジュール 2: 拡張機能の追加

このモジュールでは、拡張機能を使用して Web ページのフォーマットを変更する方法、ユーザーによるデータベースへのファイル・アップロードを可能にする機能を追加する方法、自動的にユーザーを特定のページに転送する規則をセットアップする方法、および一部の管理タスク（鍵の生成など）を自動化する方法について学習します。

学習目標

このモジュールでは、データベースのデータを、より効果的に使用する方法を学習します。このモジュールでは、以下に挙げることを行います。

- Web ページでデータベース・レコードをフォーマットする
- Web ページからデータベースへのファイルのアップロードを可能にするコンポーネントを追加する
- 自動的にページ間をナビゲートする
- データベース管理タスクの一部を自動化する

所要時間

このモジュールを完了するには、約 1 時間 30 分 を要します。チュートリアルを学習しながら動的 Web サービスの他の面を調べるように決めた場合は、終了するのにこれより長くなる場合があります。

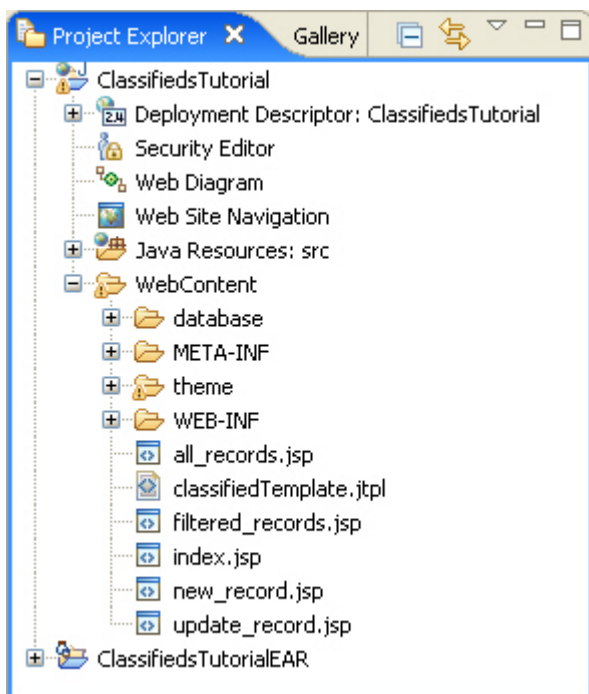
前提条件

『モジュール 1: データ接続を扱う Web ページの作成』をすでに完了している場合は、この前提条件セクションの残りの部分をスキップして、32 ページの『演習 2.1: データ・テーブルのフォーマット』の作業を開始してください。

このチュートリアルを (モジュール 1 を行わずに) モジュール 2 から始める場合には、最初に必須リソースのインポート、ターゲット・サーバーのセットアップ、およびデータベース接続のセットアップを行う必要があります。

必須リソースをインポートする手順は、次のとおりです。

1. プロジェクトをインポートします。Web パースペクティブに切り替えます (「ウィンドウ」 → 「パースペクティブを開く」 → 「Web」)。
2. Web パースペクティブの「プロジェクト・エクスプローラー」ビューで、使用する ClassifiedsTutorial プロジェクトが次のようなイメージになっていることを確認します。



ターゲット・サーバーを設定する手順は、次のとおりです。

1. Web パースペクティブの「プロジェクト・エクスプローラー」ビューで、「ClassifiedsTutorial」を右クリックして「プロパティ」を選択します。

2. プロパティ・リストで、「**サーバー**」をクリックします。
3. デフォルト・サーバー・リストで、デフォルトとして使用するサーバーを選択します。「**適用**」をクリックします。
4. プロパティ・リストで、「**ターゲット・ランタイム**」をクリックします。
5. ランタイム・リストで、選択したサーバーに対応するランタイムをクリックします。「**OK**」をクリックします。

使用したいサーバーがターゲット・ランタイム・リストにない場合は、ワークベンチを閉じて該当サーバーをインストールしてください。サーバーのインストールが完了したら、説明に従ってターゲット・サーバーを設定します。

注: サーバー・ランタイムをインストール済みであるのにデフォルト・サーバー・リストにサーバーが一切表示されない場合は、サーバーを構成する必要があると考えられます。サーバーは、次の手順で構成します。

1. 「**all_records.jsp**」ファイルを右クリックして、「**実行**」 → 「**サーバーで実行**」をクリックして選択します。
2. 「**手動で新規サーバーを定義**」を選択します。
3. インストールしたサーバーを選択します。
4. ウィザードの指示に従って、サーバーを構成します。サーバーで初めて実行するときに、エラーを受け取る場合があります。エラーを修正するには、上記のとおりターゲット・サーバーを設定し、「サーバー」ビューでサーバーを再始動してから、ブラウザーに Web ページを再ロードします。

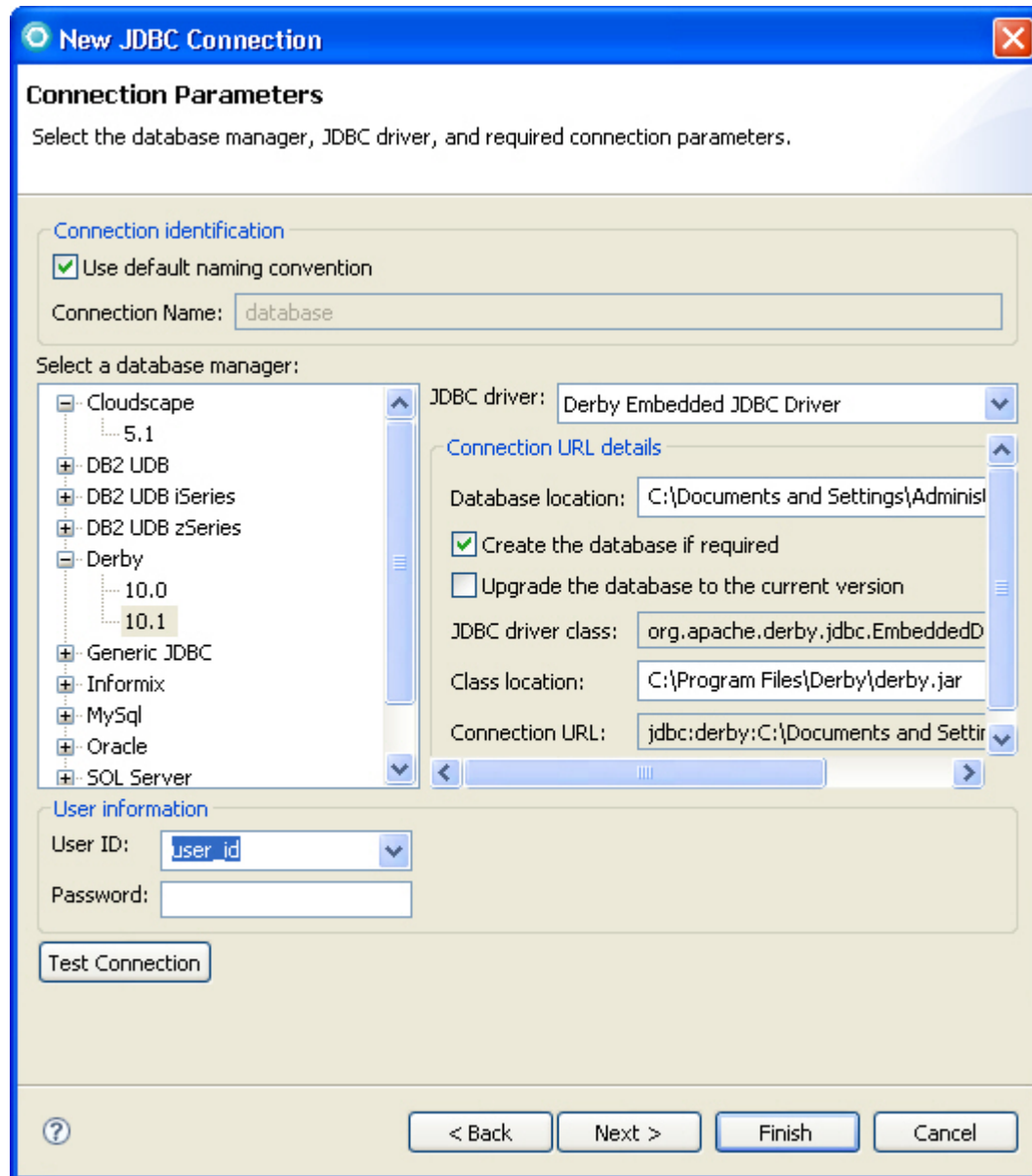
前述のターゲット・サーバー設定のステップを繰り返すと、ここで構成したサーバーがデフォルト・サーバーになっているはずです。インストール済みであっても未構成のサーバーは、デフォルト・ターゲットを選択するためのサーバーのリストに表示されません。

WebSphere Application Servers 以外のサーバー上への SDO のデプロイについては、『ヘルプ・トピック: WAS サーバー以外のサーバーへの SDO のデプロイ (Deploying SDOs on non-WAS servers)』を参照してください。

データベース接続をセットアップする手順は、次のとおりです。

1. 「プロジェクト・エクスプローラー」ビューで、「**ClassifiedsTutorial**」を右クリックして「**プロパティ**」を選択する。「ClassifiedsTutorial のプロパティ (Properties for the ClassifiedsTutorial)」ウィンドウが開きます。
2. 「**JDBC 接続**」をクリックする。
3. 「**JDBC 接続**」プロパティで、「**新規**」をクリックする。「**新規 JDBC 接続**」ウィンドウが開きます。
4. 「**新規接続の作成**」をクリックし、「**次へ**」をクリックする。
5. 「**データベース・マネージャーの選択**」リストで、「**Derby**」を展開し、次に「**10.1**」を選択する。
6. 「データベース・ロケーション」フィールドで、「**参照**」をクリックし、「**<workspace_location>¥ClassifiedsTutorial¥WebContent¥¥database**」を選択する。ここで、「**<workspace_location>**」は現行ワークスペースのディレクトリです。「**OK**」をクリックして、「**フォルダーの参照**」ウィンドウを閉じます。
7. 「**クラス・ロケーション**」フィールドで、JDBC ドライバー・クラスを実装する JAR (複数可) のロケーションを入力するか、「**参照**」をクリックしてファイル・システムからロケーションを選択する。JAR ファイルのロケーションを選択したら、「**開く**」をクリックします。

8. データベースにアクセスするには、ユーザー ID の入力が必要となる場合があります。パスワードは必要ありません。「新規データベース接続」ウィザードは、次のようになるはずです。



9. 「新規 JDBC 接続」ウィザードで、「終了」をクリックする。「ClassifiedsTutorial のプロパティー (Properties for ClassifiedsTutorial)」ウィンドウに戻ります。
10. 「プロジェクトのデータベース接続の構成」をクリックする。「JDBC 接続プロパティー (JDBC Connections properties)」ウィンドウが開きます。
11. 「ランタイム接続の詳細」セクションの隣にある「編集」をクリックする。ランタイム接続のタイプとして「ドライバ・マネージャ接続の使用」を選択する。「終了」をクリックして、次に「OK」をクリックする。

チュートリアル「Web プロジェクト」のファイルを参照することができます。ファイルを開くには、「プロジェクト・エクスプローラー」ビューでそのファイルをダブルクリックします。各ページの関連状況をマップ形式で表示するには、プロジェクト・エクスプローラーで「Web サイト・ナビゲーション」をダブルクリックします。

このサンプルで行う大部分の作業は次のファイルに関連しています。

all_records.jsp

これはサイトのホーム・ページです。データベースにあるすべての案内広告を表示します。

new_record.jsp

このページは新規案内広告を作成します。

update_record.jsp

このページはデータベースにある広告に関する詳細を変更するか、または削除します。

classifiedTemplate.jtpl

これはサイトのページのためのテンプレートです。これには、テーブルや、すべてのページにあるグレーの「広告へようこそ」バナーなどの要素が含まれています。このページには、グレーのバナーの下に 2 つのナビゲーション・タブもあり、ホーム・ページと新規案内広告ページに進むことができます。

演習 2.1: データ・テーブルのフォーマット

all_records.jsp ページにあるデータ・テーブルは、実用的な外観を持っています。これはデータベースのすべてのレコードを表示して、意図されたとおりに他のページにリンクします。ただし、見栄えがするものとはいえません。この演習では、このデータ・テーブルにページャー、スタイル規則、および販売品目の写真を追加することによってデータ・テーブルを改善します。

列ヘッダーの名前変更

多くの場合、ユーザーのデータ・テーブルの列ヘッダーをデータベースの列名と全く同じにしたいとは思わないでしょう。以下の手順で、データ・テーブルの列ヘッダーをより適切なものに名前変更する方法を示します。

1. 「プロジェクト・エクスプローラー」ビューで、「**all_records.jsp**」ページをダブルクリックします。
2. データ・テーブルの「**MAINCATEGORY**」列ヘッダーをクリックします。
3. 「プロパティ」ビューで「**h:outputText**」タブをクリックし、「値」の 1 行テキスト領域を「**MAINCATEGORY**」から「**CATEGORY**」に変更します。
4. 必要であれば、その他の列も名前変更します。
5. このページを保管します。

出力コンポーネントのフォーマット

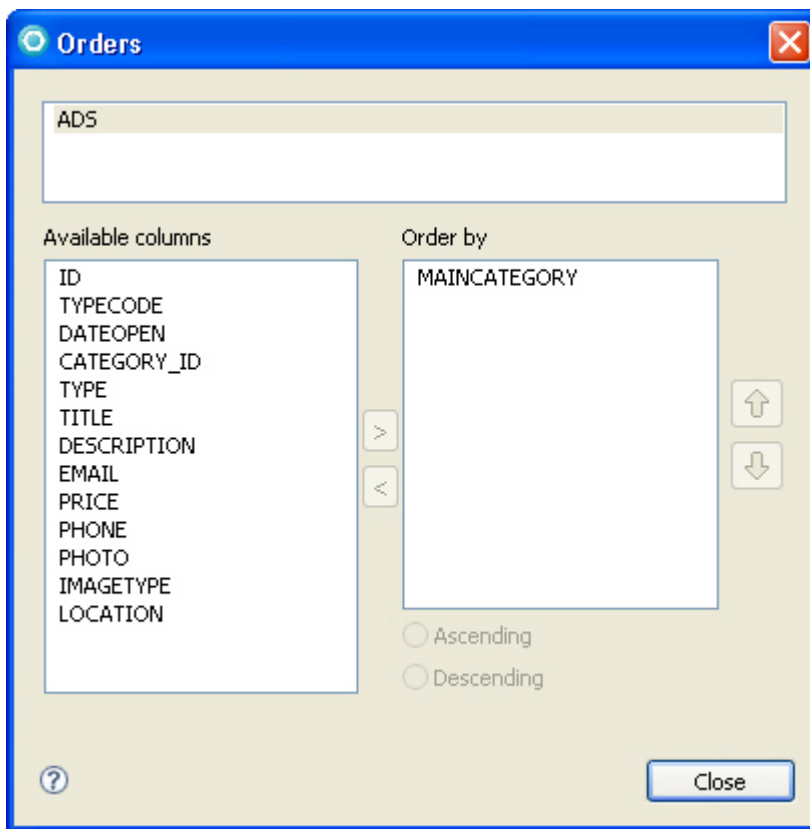
出力コンポーネントもフォーマットできます。以下の手順では、{PRICE} 出力コンポーネントを、通常の数値としてではなく通貨として表現されるようにフォーマットします。

1. ページの {PRICE} 出力コンポーネントをクリックします。
2. 「プロパティ」ビューで「型」ドロップダウン・リストを見つけ、「通貨」を選択します。これで、各品目の価格が、通常の数値ではなく通貨スタイルで表示されるようになります。さまざまな型のデータ (例えば、パーセント、日付、時刻など) には、別のスタイルおよびフォーマットが使用可能です。さらに、出力コンポーネントのフォーマットをカスタマイズして、電話番号のような「プロパティ」ビューにリストされていないデータの型をカスタマイズすることもできます。
3. このページを保管します。

データのソート

この時点では、データのリストはどのような順序にも並んでいません。より整理された広告のリストをユーザーに提供するには、類似品目がグループ化されるようにカテゴリー別にレコード・リストを順序付けます。

1. 「ページ・データ」ビュー（「ウィンドウ」 → 「ビューの表示」 → 「ページ・データ」）の「all_recordlist (ADS)」を右クリックして、コンテキスト・メニューの「構成」をクリックします。
「関連レコード・リストの構成」ウィンドウが開きます。使用するデータベースへの接続が確立できなかったことを示す警告メッセージが表示された場合は、Web サイトのテスト後、サーバーを実行したままにしておいたのです。この警告メッセージが表示された場合は、それぞれのダイアログの「キャンセル」をクリックして、『Web サイトのテスト』にあるセクション『サーバーの停止』の説明に従って、サーバーを停止してください。
2. 「スコープから既存のレコードまたはレコード・リストを取得」と「既存のレコードまたはレコード・リストからメタデータ定義を再利用」を選択し、「次へ」をクリックします。
3. 次のウィンドウで「ADS」をクリックして選択し、「次へ」をクリックします。
4. 次のウィンドウの右側にある「配列結果」をクリックします。その結果、「配列」ウィンドウが開きます。
5. 「使用可能な列」ペインで、「MAINCATEGORY」をクリックします。
6. 「>」ボタンをクリックして、「MAINCATEGORY」列を「配列基準」ペインに移動します。ウィンドウは次のようになります。



7. 「閉じる」をクリックして、「配列」ウィンドウを閉じます。
8. 「終了」をクリックして、変更を適用します。これで、データはカテゴリー順に並び替えられました。データベース内の任意の列でソートを行うことができます。

ページャーの追加

対象のデータ・テーブルにレコードすべてを一度に表示するのではなく、ページャーを使用することができます。ページャーは、プロジェクト内で新規 JSP ファイルを作成せずに、ページに表示するレコードを特定のサイズに自動的に分割します。

1. データ・テーブル内の任意の場所をクリックします。
2. 「プロパティ」ビューで、ビューの左側にある HTML タグ・リストから「**h:dataTableEx**」をクリックします。
3. 「**h:dataTableEx**」の下「表示オプション」タブをクリックします。
4. 「ページ当たりの行数」フィールドに「5」を入力します。
5. 「**Web スタイル・ページャーの追加**」の横のボタンをクリックします。ページャーの外観を示す画像がデータ・テーブルの下部に表示されます。Page Designer では、ページャーの視覚化は、プレースホルダー・イメージであり、実際に表示されるページ数は反映しません。実際の表示ページ数は、ページがブラウザーにロードされるときにのみ決めることができます。
6. このページを保管します。必要であれば、種々のスタイルのページャーとページ統計コンポーネントをテストして、ページに最も適切なものを見つけることができます。

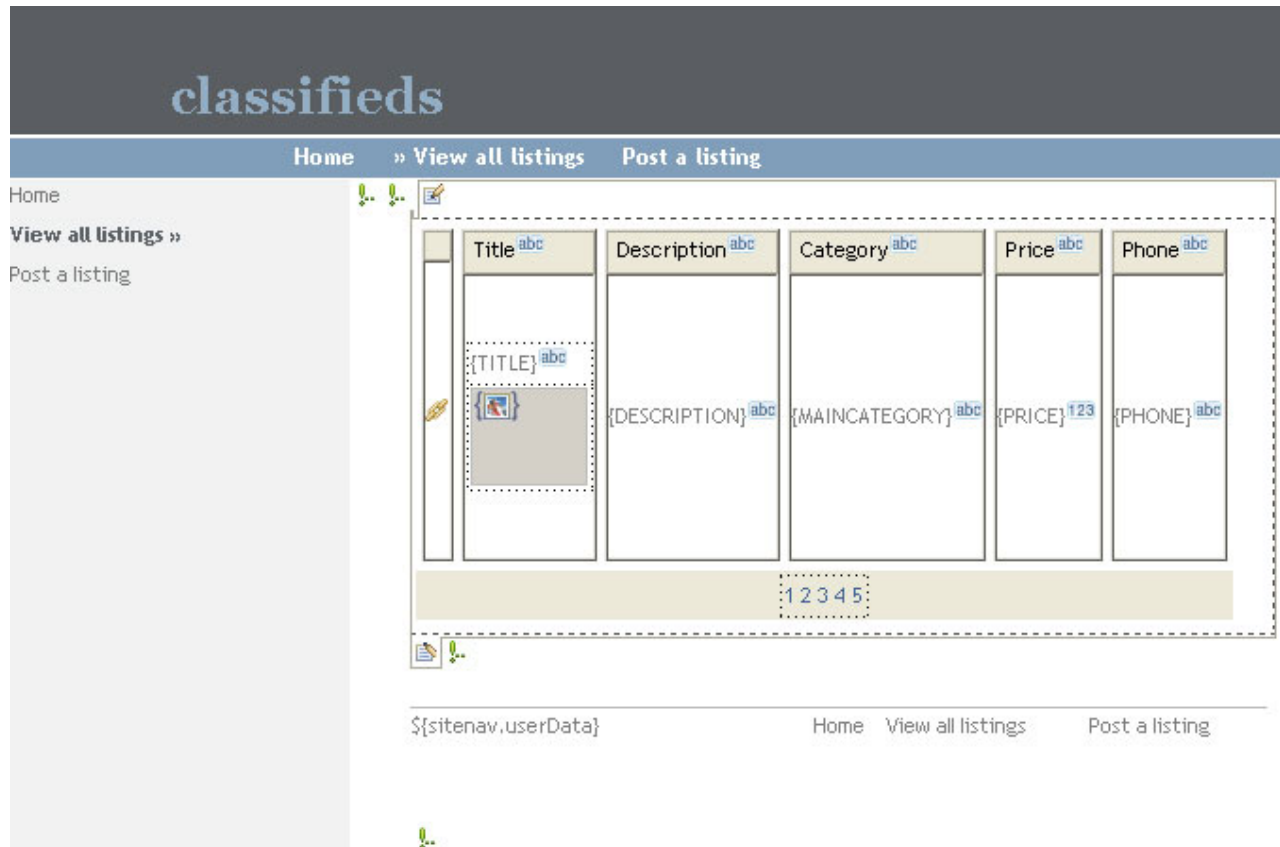
リストとしてグループ・ボックスにコンポーネントをレイアウト

データ・テーブルのそれぞれの列が常に 1 つだけ出力コンポーネントを持つのではなく、複数の列内の要素を結合して、より見栄え良くレイアウトすることが可能です。コンポーネントは、非表示 HTML 表を使用する場合と同じようにグループ・ボックスを使用して配置します。具体的には、サイトを視覚的に充実させるためにイメージ・コンポーネントを追加した後、それぞれの広告に関するほとんどのテキスト情報を取り込んで、それらを単一のデータ・テーブル列に置きます。この列は、その後、「詳細 (DETAILS)」というラベルを付けられます。

1. パレットで拡張 Faces コンポーネント・ドロワーをクリックして開きます。「**パネル - グループ・ボックス**」コンポーネントをデータ・テーブルの「TITLE」列にドラッグします。「データ型の選択」ウィンドウが開きます。
2. 「データ型の選択」ウィンドウで、グループ・ボックス型として「**ボックス**」を選択して、「**OK**」をクリックします。このグループ・ボックスは、その中に置かれたコンポーネントを、単一の列または行にレイアウトします。このリストは、垂直または水平のいずれかの向きにすることができます。
3. 「グループ・ボックス」をクリックして選択します。
4. 「プロパティ」ビューを使用して、「**向き**」を「**垂直方向**」に変更します。
5. **{TITLE}** コンポーネントをリスト・グループ・ボックスにドラッグします。コンポーネントをグループ・ボックスに追加すると、このボックスの説明テキストはなくなります。
6. 「**イメージ**」コンポーネントを、パレットの拡張 Faces コンポーネント・ドロワーからリスト・グループ・ボックスにドラッグします。この操作は、イメージ・コンポーネントをグループ・ボックスの下辺にドロップして、イメージを確実にタイトル以下に配置するのに役立ちます。
7. 新たに追加したイメージ・コンポーネントをクリックします。
8. 「プロパティ」ビューの「**サイズ**」エリアを使用して、「**幅**」を 60 ピクセル、「**高さ**」を 50 ピクセルに設定します。こうすることで、これらのイメージは、データベースにおけるイメージのサイズに関係なく、常に 60x50 サイズでページに表示されるようになります。
9. 「ページ・データ」ビューから「**写真 (PHOTO)**」列をイメージ・コンポーネントにドラッグして、このイメージ・コンポーネントを all_recordlist の「**写真 (PHOTO)**」列にバインドします。これにより、イメージ・コンポーネントは、それぞれのレコードの「**写真 (PHOTO)**」列で検出されたイメージ・データを表示します。「**写真 (PHOTO) (ByteArray)**」列がリストされていない場合は、「ページ・データ」ビューの「**all_recordlist (ADS)**」を右クリックして、メニューの「**構成**」を選択します。「**OK**」をクリックして、「**URL**」ウィンドウが開いている場合は閉じます。「**関連レコード・リストの構成**」ウィンドウで、「**次へ**」をクリックします。「**ADS**」を選択して「**次へ**」をクリックします。「**写真 (PHOTO) (ByteArray)**」の隣にあるボックスをクリックしてチェック・マークを付け、「**終了**」をクリックします。

10. 挿入したグラフィックをクリックします。「プロパティ」ビューの「**hx:graphicImageEx**」タブで、「型」フィールドの隣にある「**ページ・データ・オブジェクトの選択**」  ボタンをクリックします。「ページ・データ・オブジェクトの選択」ウィンドウが開きます。
11. 「**all_recordlist (サービス・データ・オブジェクト)**」 → 「**all_recordlist (ADS)**」をクリックして展開します。
12. 「**イメージ・タイプ (ストリング) (IMAGETYPE (string))**」をクリックします。
13. 「**OK**」をクリックして、ページを保管します。

このページは、次のようになるはずです。

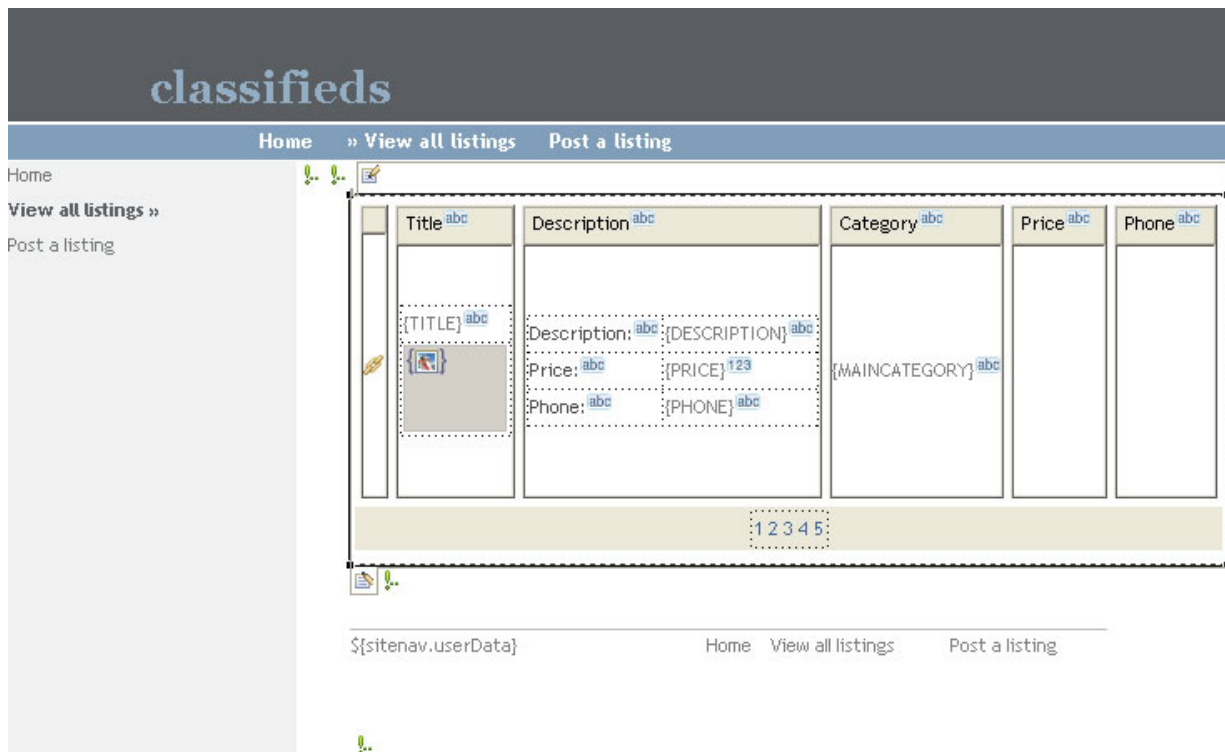


グループ・ボックス・グリッドを使用したレイアウトの調整

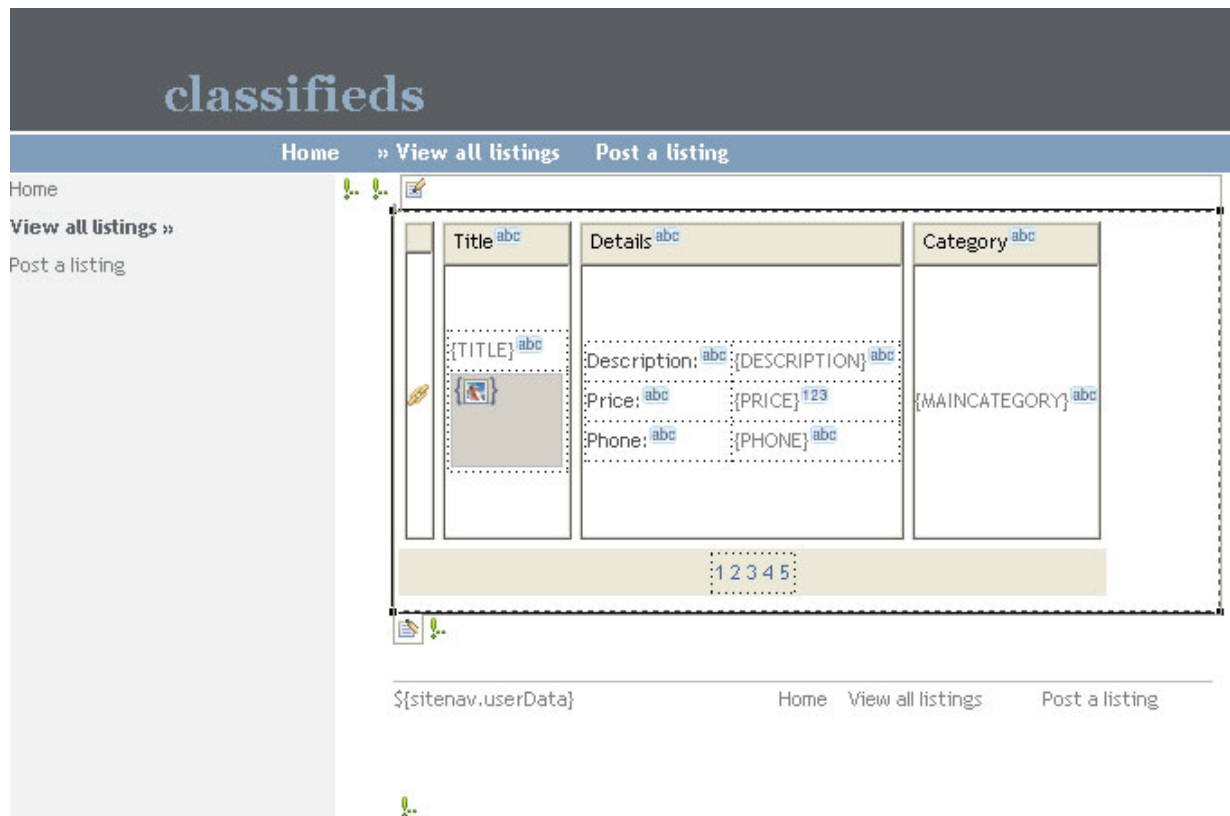
グループ・ボックスは、テーブルのようなフォーマットでコンポーネントを編成することもできます。リスト型のグループ・ボックスは行または列を 1 つだけ持つことができますが、グリッド型のグループ・ボックスは任意の数の行および列を持つことができます。以下の手順で、{PRICE} および {PHONE} コンポーネントをデータ・テーブルの「DESCRIPTION」列に移動し、それらのためのラベルを組み込みます。

1. パレットの拡張 Faces コンポーネント・ドロワーから、「**パネル - グループ・ボックス**」コンポーネントをデータ・テーブルの「DESCRIPTION」列にドラッグします。「データ型の選択」ウィンドウが開きます。
2. 追加するコンポーネントの型として「**グリッド**」をクリックし、「**OK**」をクリックします。
3. 新規グリッドのグループ・ボックスをクリックする。
4. 「プロパティ」ビューを使用して、このグループ・ボックスの「**列**」数を「2」に設定します。

5. パレットの Faces コンポーネント・ドロワーから、「出力」コンポーネントをグリッドの「グループ・ボックス」にドロップします。 この出力コンポーネントは販売品目の説明のラベルになります。それぞれの出力コンポーネントには、テーブルの左側のセルで上述のようなラベルが付けられます。
6. 「出力」コンポーネント (Page Designer に「outputText」として表示される) をクリックし、「プロパティ」ビューを使用して、それに「説明:」の「値」を割り当てます。
7. 既存の {DESCRIPTION} コンポーネントをグリッドの「グループ・ボックス」の下辺にドラッグします。 コンポーネントをこのグループ・ボックスの正しい場所にドラッグしにくい場合は、マウス・ボタンを押したままにして、カーソルを監視してください。グループ・ボックス内のカーソルの位置は、マウス・ボタンを放したときにコンポーネントが現れる場所を示しています。この演習では、カーソルが直前のコンポーネントの右側に来たときに、マウス・ボタンを放してください。
8. パレットから別の「出力」コンポーネントをドラッグして、「グループ・ボックス」の右側にドロップします。
9. 「プロパティ」ビューを使用して、それに「価格:」の「値」を割り当てます。
10. 既存の {PRICE} 出力コンポーネントを「価格」出力テキストの右側にドラッグします。
11. 同様の方法で、{PHONE} コンポーネントのラベル用として新しい「出力」コンポーネントをドラッグし、それに「電話:」というラベルを付けます。
12. 既存の {PHONE} コンポーネントを、「電話」出力テキストの右にあるグリッドのグループ・ボックスにドラッグします。 このページは、次のようになるはずです。



13. データ・テーブルの空の「PRICE」と「PHONE」の列を削除します。列を削除するには、該当する列をクリックして、「プロパティ」ビューを開きます。次に、ビューの左側にある HTML タグ・リストから **h:dataTable** をクリックして、ビューの右側にあるリストから削除したい列を選択し、「除去」をクリックします。
14. 「説明」列のヘッダーをクリックして、「プロパティ」ビューを使用し、その「値」フィールドを「詳細」に変更します。 このページは、次のようになるはずです。



15. このページを保管してテストする。サーバーでのアプリケーション実行の経験が豊富でない場合は、14 ページの『演習 1.3: Web サイトのテスト』を参照してください。

このページをテストするときの外観は、次のようになるはずです。

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

関連イメージが入っていない広告では、壊れたリンクが表示されます。 次の演習では、ファイル・アップロード・コンポーネントをページに追加して、広告のイメージを追加または変更する方法を学習します。

演習のチェックポイント

演習 2.1 が完了しました。この演習では、データ・テーブルをさまざまな方法でフォーマットする方法を学習しました。

次のことを学習しました。

- 列見出しを変更する。
- テーブル内のデータをソートする。
- ページャーを追加してデータをさらに小さいいくつかの Web ページに分割する。
- さまざまな方法でグループ・ボックスを使用し、Web ページ上のテーブルのレイアウトを変更する。

演習 2.2: ファイル・アップロード・コンポーネントの使用

この演習から、ユーザーが案内広告用ピクチャーをアップロードできるようにする方法を学習します。この機能は `new_record.jsp` および `update_record.jsp` のページでは、特に重要です。ファイル・アップロード・コンポーネントにより、ユーザーは自分のファイル・システムを参照し、ファイルをデータベースにアップロードして、ただちに、このファイルをデータベースで見えるようにすることができます。

ファイル・アップロード・コンポーネントを使用する手順は、作成と更新の両方の場合で類似しているのので、この演習では、更新ページで任意のリストのための現在のイメージを変更できます。お望みであれば、更新ページの使用を終了するときに、新規レコード・ページに同じ変更を行うことができます。

現在の写真をページへ追加

更新フォームは、静的テキストと「`update_record`」関連レコードの列にバインドされた入力コンポーネントを持つ HTML 表と何ら変わらないことに気がつくでしょう。このことが分かれば、行と列を追加して、HTML 表の変更とちょうど同じようにフォームを変更することができます。以下の手順で、現在の写真を表示する新しい行を追加します。

行と列を追加することは、関連レコード・リストを表示するデータ・テーブルでは、かなり複雑です。列をデータ・テーブルに追加するための情報については、7 ページの『演習 1.2: データベースへの接続と Web ページ上でのデータの表示』を参照してください。

1. 「プロジェクト・エクスプローラー」ビューで、「`update_record.jsp`」ページをダブルクリックする。
2. カーソルをテーブルの最初の (左上部の) セルに置く。このセルはラベル付きの ID です。
3. メインメニューから「表」→「指定して追加」をクリックします。「行または列の追加」ダイアログが開きます。「行」をクリックし、次に「選択されたセルの上」をクリックします。これにより、レコード用の現在の写真が入ることになるテーブルの上部に新規行が作成されます。「OK」をクリックします。
4. この新しい行の左セルに、ラベルとして「現在の写真: (Current photo:)」と入力します。
5. パレットの Faces コンポーネント・ドロワーから、「イメージ」コンポーネントを新規行の右端のセルにドラッグする。直前の演習とは違って、ユーザーにはサイズの制約なしでフル・イメージを見ることを許可します。したがって、「プロパティ」ビューで幅と高さを変更しないでください。
6. 「ページ・データ」ビューから「写真 (PHOTO)」列を新規イメージ・コンポーネントにドラッグして、このイメージ・コンポーネントを `update_record` の「写真 (PHOTO)」列にバインドする。これで、イメージ・コンポーネントが、データベースの写真 (PHOTO) 列にバインドされました。
7. グラフィック・イメージを選択した状態で、「プロパティ」ビューに進み、「型」フィールドの隣にある「ページ・データ・オブジェクトの選択」ボタンをクリックする。「ページ・データ・オブジェクトの選択」ウィンドウが開きます。
8. 「`update_record` (サービス・データ・オブジェクト)」→「`update_record` (ADS)」→「イメージ・タイプ (ストリング) (IMAGETYPE (String))」をクリックします。
9. 「OK」をクリックすると、写真がある場合は、ページに案内広告のための現在の写真が表示されます。このページは、次のようになるはずです。

classifieds

[Home](#)
[View all listings](#)
[Post a listing](#)

[Home](#)
[View all listings](#)
[Post a listing](#)

Current photo:



Id:

{ID}123

Title:

{TITLE}abc

Description:

{DESCRIPTION}abc

Maincategory:

{MAINCATEGORY}abc

Phone:

{PHONE}abc

Price:

{PRICE}123

{Error Messages}

Update

Delete

{sitenav.userData}

[Home](#)
[View all listings](#)
[Post a listing](#)

ページへのファイル・アップロード・コンポーネントの追加

次に、新規行をテーブルの下部に追加して、ファイル・アップロード・コンポーネントを入れます。

- 最終行をクリックして、カーソルを最終行に置く。
- メニュー・バーで、「表」 → 「行を下へ追加」とクリックする。
- 新規行の最初のセルに、ラベルとして「新規の写真: (New photo)」と入力する。
- パレットの拡張 Faces コンポーネント・ドロワーから、「ファイル・アップロード」コンポーネントを新しい行の最終セルにドラッグする。
- イメージ・コンポーネントのバインドと同じ方法で、ファイル・アップロード・コンポーネントを「update_record」関連レコードの「写真 (PHOTO)」列にバインドする。「{PHOTO}」が 1 行テキスト領域に表示されて、このコンポーネントが「写真 (PHOTO)」列にバインドされていること、およびアップロードされるファイルがこの列に置かれることを示します。このページは、次のようになるはず

規則のセットアップ

1. プロジェクト・ナビゲーターで「**new_record.jsp**」ページをダブルクリックする。
2. そのページの「新規リストの掲載 (Post New Listing)」ボタンをクリックする。
3. 「プロパティー」ビューで「規則の追加」をクリックする。「ナビゲーション規則の追加」ウィンドウが開きます。新規レコードをデータベースに作成してコミットしたときに何か間違いがあれば、最初の規則によって、ユーザーは `create_error.jsp` という名前のエラー・ページに送られます。
4. 「ページ」フィールドに、「`error_create.jsp`」と入力する。このページは実在しませんが、このチュートリアルの上、存在するものと想定してください。
5. 「指定された結果」ラジオ・ボタンをクリックする。
6. 「指定された結果」ラジオ・ボタンの後にある 1 行テキスト領域に「`ERROR_CREATE`」と入力する。
7. 「このページのみ」ラジオ・ボタンをクリックする。このサイトには、ユーザーが重複 ID 番号を入力してこの特定エラーのトリガーとなる可能性のある別のページがないので、このラジオ・ボタンを使用します。「**OK**」をクリックします。次の規則は、ユーザーが有効な ID 番号を入力した場合に、`all_records.jsp` に戻るようナビゲートするものです。
8. 「規則の追加」ボタンをクリックして、再度、「ナビゲーション規則の追加」ウィンドウを開く。
9. 「ページ」ドロップダウン・ボックスを使用して、「**all_records.jsp**」を選択する。
10. 「指定された結果」チェック・ボックスをクリックし、それに続く 1 行テキスト領域に「`MAIN`」と入力する。
11. 別のページ (例えば、`update_record.jsp` ページ) でこの規則を再利用する必要がある場合には、「この規則の使用対象」の下に「すべてのページ」ラジオ・ボタンをクリックする。「**OK**」をクリックします。これで、この 2 つの規則は、「プロパティー」ビューに表示されます。

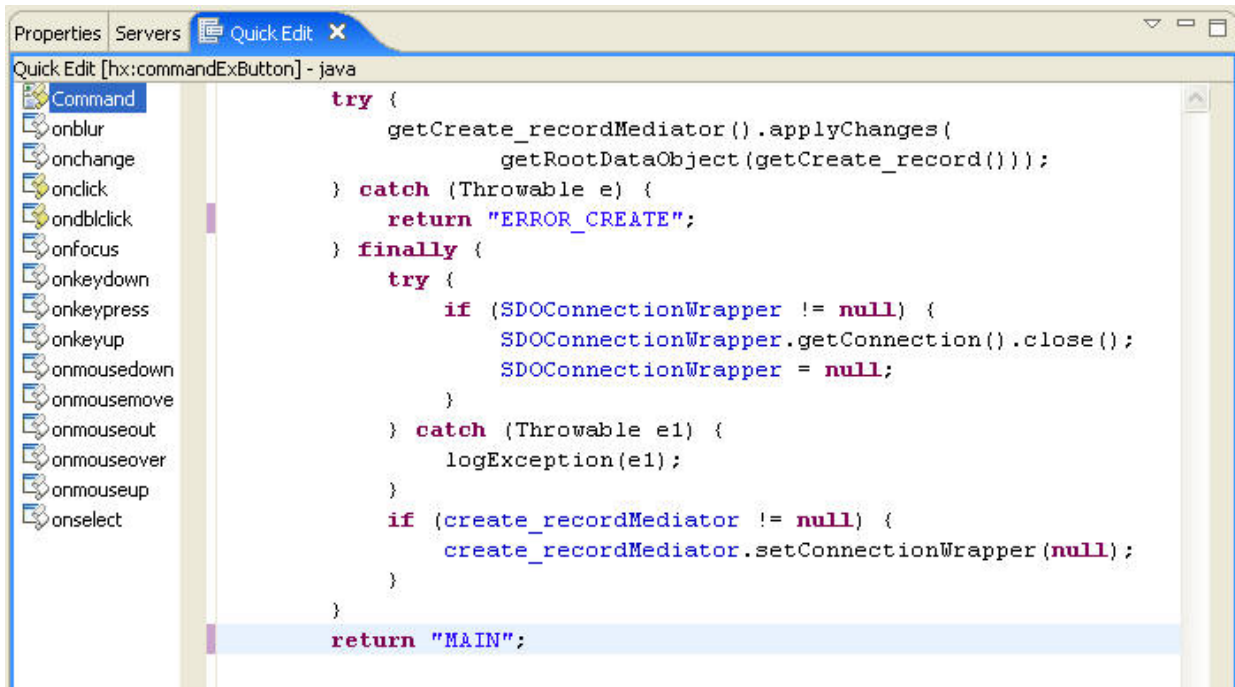
ボタン・アクションから別名を戻す

現在、すべきことで残されているすべては、使用する新規ナビゲーション規則が働くようにすることだけです。「新規リストの掲載 (Post New Listing)」ボタンのためのコードには、2 つの「`return`」文を追加します。これらの「`return`」文は、ユーザーがナビゲーション規則で決まっている適切なページに送られるように、アプリケーション別名を呼び出します。

1. 「新規リストの掲載 (Post New Listing)」ボタンをクリックする。
2. 「クイック編集」ビューを開く。
3. 「クイック編集」ビューで、「`} catch (Throwable e) {`」という行を見つける。この `catch` 関数は、ユーザーが重複 ID 番号を入力した場合に実行されます。
4. この行の終わりの開き中括弧「`{`」と 2、3 行下にある次の閉じ中括弧「`}`」で囲まれているすべてのコードを除去する。その際には、どちらの中括弧も除去しないでください。
5. 除去したコードの代わりに、これらの中括弧の間にテキスト「`return "ERROR_CREATE";`」を入力する。
6. 次に、最後の閉じ中括弧 `}` 以降にあるすべてのコードを除去し、代わりにテキスト「`return "MAIN";`」を入力する。

このステップによって、演習 1.4 で追加した「ページヘジャンプ」アクションが除去されます。このコードは、ナビゲーション規則が同じことを行うので、不要になりました。

ユーザーのボタンのコードは、次のようになるはずです。



7. このページを保管し、必要に応じてテストする。

オプションで、ユーザーがリストを作成するときにエラーが発生したので、異なる ID 値を試行するようにとユーザーに説明する、create_error.jsp という名前の単純なエラー・ページを作成することができます。ここで、既存の ID 値 (例えば「1」) を使用する新規リストの追加を試みることによって、これらのナビゲーション規則をテストしてみることができます。

演習のチェックポイント

演習 2.3 が完了しました。この演習では、ナビゲーション規則を使用して、ユーザーが無効な ID の値を入力しようとした場合にエラーを作成する方法、または有効な ID の値を入力したユーザーを all_recordlist.jsp に転送する方法を学習しました。

演習 2.4: 自動キー生成の使用

直前の演習では、新しい案内広告のためにユーザーが固有の ID を入力したことを確認するためのナビゲーション規則をセットアップしました。サイト・ユーザーに試行錯誤させて固有の ID を探し当てさせたくないため、このプロセスはもどかしく、現実的なものではありません。この演習では、データベースがデータベースのそれぞれの新規レコードに固有の番号を自動的に割り当てるように自動キー生成をセットアップします。

自動キー生成は複雑なトピックですが、簡単に言えば、キー生成のためにデータベースに特別なテーブルが予約されていると、データベースは新規キーを選択できるということです。このテーブルには、未使用キーのリストを持つ列 (増分 (Incrementor) 列) と 1 から始まる番号順のリストを持つ別の列 (識別 (Identity) 列) が必要です。データベースに新規キーが必要になると、識別 (Identity) 列で 1 を持つ行からキーを取り込み、次に、次回に使用できる新規キーを取得します。

自動キー生成についてさらに学習したい方に:

自動キー生成をセットアップするには、自動キー生成の働き方について、いくつかのことを知っている必要があります。自動キー生成を使用するには、この目的のために、データベースに 1 つのテーブルをとっておく必要があります。このテーブルには、以下の 2 つの列があります。

- 増分 (*Incrementor*) 列。使用可能なキーを保管します。データベースは、新しい固有キーが必要になると、この列からキーを取り出します。
- 識別 (*Identity*) 列。番号 1 のインスタンスを 1 つだけ持つ番号のリストです。この列は、増分 (*Incrementor*) 列から選択すべきキーをデータベースに知らせます。この識別 (*Identity*) 列はキー生成テーブルの主キーです。

データベースは、新しい固有キーが必要になると、識別 (*Identity*) 列の値が 1 の行を見つけます。この行の増分 (*Incrementor*) 列の値が次に使用可能なキーを保持しています。データベースは、次回に新しいキーが使用可能になるように、このキーを使用してテーブルを更新します。

以下はキー生成テーブルの例です。このデータベースの次回に使用可能なキーは何でしょうか。答えは、このテーブルの下にあります。

表 1. キー生成テーブル

識別 (<i>Identity</i>) 列	増分 (<i>Incrementor</i>) 列
3	78
4	3
1	65
2	12

このテーブルの次回に使用可能なキーは 65 です。これは、このキー (増分 (*Incrementor*) 列にある) が、識別 (*Identity*) 列内で 1 を持つ行と同じ行にあるからです。

次回に使用可能なキーがテーブルから取り出されると、キーが必要になる次回に備えてテーブルが更新されます。データベースは、複数行の増分 (*Incrementor*) 列の値を取り出すことによって、複数のキーを 1 度に取得することもできます。

要約すると、自動キー生成が正しく機能するには、2 つの列 (識別列として使用する主キー列と、次回使用可能なキーを保管するための列) を持つキー生成テーブルを準備するだけで済みます。このテーブルは、識別列の値を 1 とし、使用する最初の使用可能なキーをその増分列の値として持つ 1 つのレコードで初期化されている必要があります。このセットアップを行えば、自動キー生成は準備完了です。

自動キー生成のセットアップ

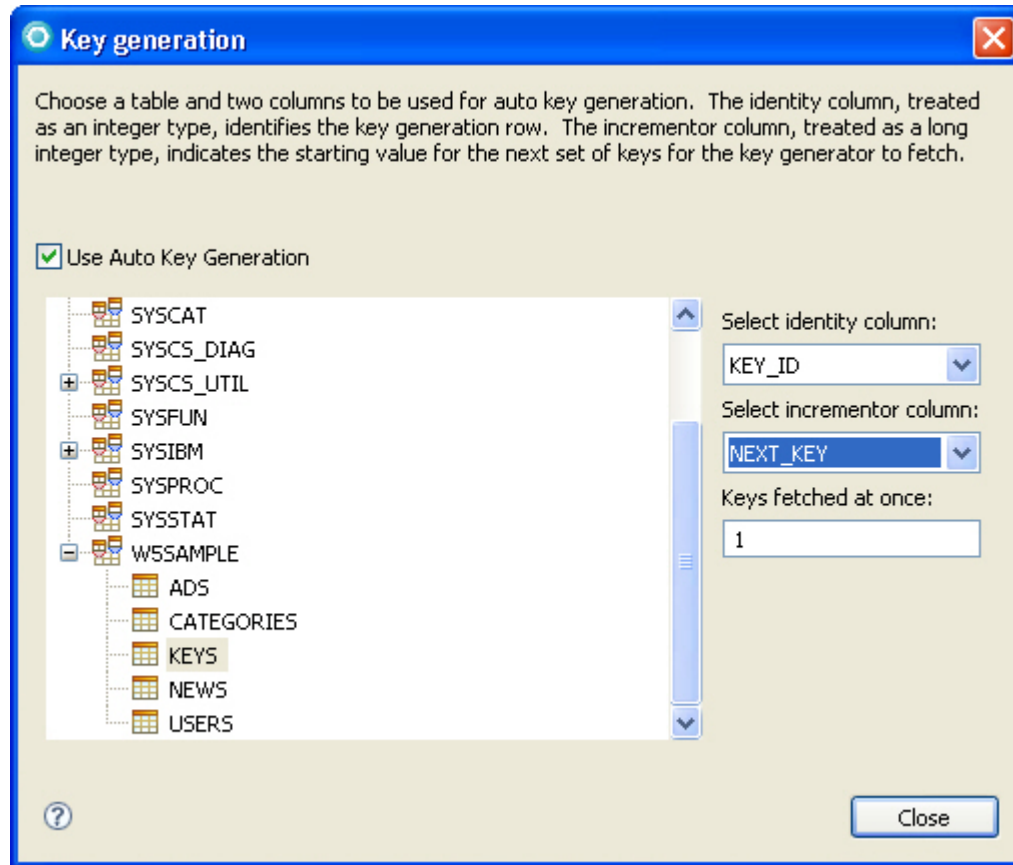
このチュートリアルで提供されたサンプル・データベースは、KEYS という名前のキー生成テーブルを持っています。上述した 2 つの列は、各新規レコードに対して新規 ID 番号を提供します。以下の手順によって、「create_record」関連レコードをセットアップし、KEYS テーブルからその ID 番号を取得します。

1. 「プロジェクト・エクスプローラー」ビューで「new_record.jsp」ページをダブルクリックする。
2. 「ページ・データ」ビューで、「create_record」関連レコードをダブルクリックする。「関連レコードの構成」ウィンドウが開きます。
3. 「既存のレコードまたはレコード・リストからメタデータ定義を再利用」および「レコードにデータベースからの既存のデータを入れてください」をクリックして、「次へ」をクリックする。
4. ドロップダウン・リストからデータベース接続を選択し、「次へ」をクリックする。
5. 「関連レコードの構成」ウィンドウの右側に、リンクのリストがあります。「自動生成キー」をクリックします。「キー生成」ウィンドウが開きます。

6. 「自動キー生成の使用」を選択する。
7. W5SAMPLE リストをクリックして展開し、「KEYS」テーブルをクリックする。
8. 「ID 列の選択」フィールドで、「KEY_ID」をクリックする。
9. 「増分列の選択」フィールドで、「NEXT_KEY」をクリックする。

広告の ID にはキーは 1 つだけ必要なので、「一度に取り出すキー数」を設定値 1 のままにしておきます。複数のキーが必要な場合は、この設定ですべてのキーを一度にデータベースに選択させることができます。

「データ・オブジェクトの構成」ウィンドウは、次のようになるはずです。



10. 「閉じる」をクリックして「キー生成」ウィンドウを閉じ、「終了」をクリックして変更を適用する。これで、各新規レコードに ID フィールドが自動的に生成されます。次に、ユーザーが値を入力できないように、ID 入力フィールドを除去する必要があります。
11. **Id:** テキストをクリックして、カーソルを入力フォーム・テーブルの先頭行に置く。
12. 「テーブル」→「削除」→「行の削除」と順にクリックする。
13. このページを保管する。

オプションとして、どんなキーが生成されたかを知りたいのであれば、行を削除する代わりに、その ID 番号の入力コンポーネントを削除し、それを create_record の ID 列にバインドされている出力コンポーネントで置換することもできます。こうすると、自動生成キーはフォームの上部に表示されますが、ユーザーはこれを変更することはできません。

完了したチュートリアルへのテスト

Web アプリケーションを公開する準備が完了したら、それをホストして、ユーザーがインターネット経由で Web サイトにアクセスできるようにするサーバーが必要になります。ただし、Web サイトをテストするために、使用可能なランタイムを使用して、テスト目的でサーバーをシミュレートすることができます。Web サイトをテストする方法については、14 ページの『演習 1.3: Web サイトのテスト』を参照してください。

演習のチェックポイント

演習 2.4 が完了しました。この演習では、自動キー生成のセットアップ方法を学習しました。

モジュール 2: 要約

『モジュール 2: 拡張機能の追加』が完了しました。

学習した内容

次のことを学習しました。

- Web ページでデータベース・レコードをフォーマットする
- Web ページからデータベースへのファイルのアップロードを可能にするコンポーネントを追加する
- ナビゲーション規則を使用して、自動的にページ間を移動する
- 自動キー生成を使用する

追加リソース

この製品、または JavaServer Faces および JSF テクノロジーについて詳しくは、下記のリンクを参照してください。

関連情報



JavaServer Faces テクノロジー



developerWorks の JSF

JavaServer Faces を使用して動的情報を Web ページに表示する

「JavaServer Faces を使用して動的情報を Web ページに表示する」チュートリアルは、完了しました。このチュートリアルで、データベースの情報を使用できる Web ページを作成するために、JavaServer Faces をどのように使用するかが学習できました。

学習した内容

このチュートリアルを完了することにより、以下の概念とタスクについて学習しました。

- Web ページをデータベースに接続する方法
- データベースのレコードを作成、読み取り、更新、および削除できるページを作成する方法
- あるページから別のページにデータを送信する方法
- Web ページの動的コンテンツをフォーマットする方法
- JavaServer Faces および JSF コンポーネントについて

追加リソース

このチュートリアルで扱ったトピックについてさらに詳しく学習したい方は、以下のソースを検討してください。

関連情報



JavaServer Faces テクノロジー



developerWorks の JSF