Лабораторна робота.

Розробка web-додатку для управління даними

Мета: розробка web-додатку, що ефективно взаємодіє з базою у середовищі MS Visual Studio .NET.

Основні відомості

Діяльність більшості web-додатків зосереджена на витяганні, відображенні і модифікації даних. Розробники перейшли від простих клієнтських додатків з локальними базами даних до розподільних систем, заснованих на централізованих базах даних і виділених серверах. В той же час розвивалися технології доступу до даних.

Для того, щоб додаток .NET міг здійснювати взаємодію з джерелом даних, необхідно встановити з'єднання з ним. Найбільш типовим сценарієм роботи web-додатку є наступний:

– встановлюється з'єднання, відкривається підключення до бази даних;

– виконується один або декілька запитів, що здійснюють внесення змін до наборів даних джерела даних, а також вибірки даних з БД;

– здійснюється відключення від джерела даних. При цьому користувач працює з від'єднаним набором даних, переглядаючи його, виконуючи фільтрацію, вносячи зміни тощо;

– при необхідності перенесення змін з від'єднаного набору даних в БД, а також при необхідності перегляд змін, внесених в БД іншими користувачами, здійснюється підключення до джерела даних, виконуються необхідні дії, після чого проводиться відключення від БД.

Web-серверні елементи управління для роботи з даними

Ці елементи управління поділяються на два типи:

1. Елементи управління джерелами даних (SqlDataSource, AccessDataSource, ObjectDataSource, XmlDataSource, SiteMapDataSource).
2. Елементи управління відображення даних (GridView, DataList, DetailsView, FormView, Repeater, Reportviewer).

Зазвичай на сторінці розміщується один з (невізуальних) елементів управління джерел даних для зв'язку зі сховищем даних; потім додаються елементи управління для відображення даних, прив'язані до джерела. Одним з найбільш удосконаленим елементом управління для відображення даних є GridView, який також дозволяють редагувати дані.

Елементи управління джерел даних

Елемент управління	Опис
SqlDataSource	служить джерелом даних, що зберігаються в базі SQL Server. Розмістивши цей елемент на сторінці, можна маніпулювати даними SQL Serval за допомогою елементів управління, що відображають дані
AccessDataSource	як SqlDataSource, але працює з даними з бази Access
ObjectDataSource	дозволяє маніпулювати даними, які збережені в створених вами об'єктах, які можуть бути згрупованими в клас колекції. Це може бути швидким способом представити користувацьку об'єктну модель на сторінці ASP.NET
XmlDataSource	аналогічно SqlDataSource, але працює з ієрархічними даними. Зручно використовувати разом з елементом управління TreeView (один з елементів управління, що призначений для візуалізації ієрархічних даних). Можна також при необхідності трансформувати дані XML за допомогою таблиці стилів XSL
SiteMapDataSource	дозволяє зв'язатися з ієрархічними даними карти сайту

Елементи управління для відображення даних

Елемент управління	Опис
GridView	відображає елементи даних (такі як записи бази даних) у формі рядків, де кожний рядок складається з стовбців, які відображають поля даних. Маніпулюючи властивостями цього елементу управління, можна вибирати, сортувати і редагувати елементи даних
DataList	відображає елементи даних, де для кожного елемента можна застосовувати

Елемент управління	Опис
	довільний шаблон відображення. Як і у випадку GridView, можна вибирати, сортувати і редагувати елементи даних
DetailsView	відображає єдиний елемент даних (запис бази даних) в табличній формі, де кожний рядок даних представляє окреме поле. Дозволяє добавляти, редагувати та видаляти елементи даних
FormView	відображає єдиний елемент даних, використовуючи шаблон. Як і DetailsView, дозволяє добавляти, редагувати та видаляти елементи даних
Repeater	подібно DataList, але без можливостей вибору та редагування
ReportViewer	удосконалений елемент управління, призначений для відображення звітних даних

Загальні відомості про web-серверний елемент управління GridView

При розробці програмного забезпечення часто виникає необхідність у відображенні табличних даних. За допомогою елемента управління GridView можна відображати, змінювати та видаляти дані з різних джерел даних, в тому числі баз даних, XML-файлів і бізнес-об'єктів, що публікують дані.

Елемент управління GridView можна використовувати для виконання наступних завдань:

– автоматична прив'язка та відображення даних з елемента управління джерела даних;

– вибірка, сортування, перегляд, зміна та видалення даних з елемента управління джерела даних.

Зовнішній вигляд і поведінку елемента управління GridView можна змінювати наступними способами:

– визначати стовпці та стилі, які налаштовуються;

– використовувати шаблони для створення елементів інтерфейсу користувача (UI), які налаштовуються;

– вводити, призначений для користувача, код у функціональність елемента управління GridView за допомогою обробників подій.

Прив'язка даних до елементу управління GridView

Елемент управління GridView допускає два способи прив'язки до даних:

– з використанням властивості DataSourceID, що дозволяє прив'язати елемент управління GridView до елементу управління джерела даних. Цей підхід дозволяє елементу управління GridView скористатися наявними можливостями елемента керування джерела даних і надати вбудовану функціональність для сортування, розбиття по сторінках і оновлення даних;

– з використанням властивості DataSource, що дозволяє виконувати прив'язку до різних об'єктів, включаючи набори даних і модулі читання даних ADO.NET. При використанні цього підходу код реалізації додаткової функціональності (наприклад, сортування, розбиття по сторінках і оновлення даних) потрібно писати самостійно.

При виконанні прив'язки до джерела даних з використанням властивості DataSourceID елемент управління GridView підтримує двосторонню прив'язку даних. Крім того, що елемент управління буде відображати повернуті дані, можна включити в елементі управління автоматичну підтримку операцій оновлення і видалення прив'язаних даних.

Форматування даних, що відображаються в елементі управління GridView

Для рядків елемента управління GridView можна задати структуру, колір, шрифт і вирівнювання. Також можна задати відображення тексту та даних, що містяться в рядках. Крім цього, можна вказати порядок відображення рядків даних: у вигляді звичайних елементів, елементів що чергуються, вибраних елементів або елементів у режимі редагування. Елемент управління GridView також дозволяє задавати формат стовпців.

Зміна та видалення даних за допомогою елемента управління GridView

За замовчуванням елемент управління GridView відображає дані в режимі «тільки для читання». Проте, елемент управління також підтримує режим редагування, в якому рядок відображається за допомогою елементів управління, що допускають редагування, як TextBox або CheckBox. Елемент управління GridView також можна налаштувати на відображення кнопки Delete, яка дозволяє видалити відповідний запис з джерела даних.

Елемент управління GridView здатний автоматично виконувати операції зміни та видалення даних

над джерелом даних, до якого він прив'язаний. Це дозволяє забезпечити можливість редагування без необхідності написання додаткового коду. Разом з цим процес зміни та видалення даних можна контролювати програмним способом, наприклад, при прив'язці елемента управління GridView до елементу управління джерела даних, доступному тільки для читання.

Елементи управління введення даних, що використовуються при відображенні рядка в режимі редагування, можна налаштувати за допомогою шаблону.

Функціональність сортування в елементі управління GridView

Елемент управління GridView підтримує сортування по одному стовпцю без створення додаткового коду. Функціональність сортування елемента управління GridView можна розширити, використовуючи подію сортування та при визначенні виразу сортування.

Функціональність розбиття по сторінках в елементі управління GridView

Елемент управління GridView володіє базовою функціональністю розбиття по сторінках. Функціональність розбиття по сторінках елемента управління GridView можна розширити за допомогою властивості PagerTemplate елемента управління GridView.

Події GridView

Функціональність елемента управління GridView можна розширювати за допомогою обробників подій. Елемент управління GridView надає події, що відбуваються як перед операціями переходу та зміни, так і після них.

Загальні відомості про web-серверний елемент управління TreeView

Web-серверний елемент управління TreeView використовується для відображення ієрархічних даних, таких як зміст або каталог файлів, у вигляді дерева.

Функціональні можливості

Елемент управління TreeView підтримує такі функціональні можливості:

– автоматична прив'язка даних, завдяки якій вузли елементу управління прив'язуються до ієрархічних даних, таким як XML-документ;

– підтримка структури переходів по web-сайту завдяки інтеграції з елементом управління SiteMapDataSource;

– текст вузла, який може відображатися як звичайний текст або як гіперпосилання;

– налаштування зовнішнього вигляду за допомогою тем, визначених користувачем зображень та стилів;

– програмний доступ до об'єктної моделі TreeView, яка дозволяє динамічним чином створювати деревовидні структури, заповнювати вузли, встановлювати властивості тощо;

– заповнення клієнтських вузлів (на браузерах що підтримуються);

– можливість відображення прапорців поруч з кожним вузлом.

Елемент управління TreeView може відображати різні типи даних: статичні дані, задані декларативно в елементі управління, дані, прив'язані до елементу управління, або дані, додані до елементу управління TreeView програмним шляхом в результаті дій користувача.

Відображення статичних даних

Статичні дані в елементі управління TreeView можна відображати шляхом створення колекції елементів TreeNode, які є його дочірніми елементами. Ці дочірні елементи називаються також *дочірніми вузлами*.

Приклад, демонстрації розмітки елемента управління TreeView з трьома вузлами, деякі з яких мають дочірні вузли.

```
<asp:TreeView ID="MyTreeView" Runat="server">
  <Nodes>
    <asp:TreeNode Value="Child1" Expanded="True" Text="1">
      <asp:TreeNode Value="Grandchild1" Text="A" />
      <asp:TreeNode Value="Grandchild2" Text="B" />
    </asp:TreeNode>
    <asp:TreeNode Value="Child2" Text="2" />
    <asp:TreeNode Value="Child3" Expanded="True" Text="3">
      <asp:TreeNode Value="Grandchild1" Text="A" />
```

```
    </asp:TreeNode>
  </Nodes>
</asp:TreeView>
```

Прив'язка даних до елементу управління TreeView

Елемент управління TreeView можна прив'язати до джерела даних, який реалізує інтерфейс IHierarchicalDataSource, наприклад, елементи управління XmlDataSource і SiteMapDataSource. Крім того, при прив'язці даних можна керувати тим, які поля слід заповнювати з джерела даних.

Відображення даних програмним шляхом за допомогою TreeNodeCollection

Елемент управління TreeView можна заповнити даними програмним шляхом, звернувшись до властивості Nodes, яку повертає клас TreeNodeCollection. Колекція TreeNodeCollection є строго типізованою колекцією об'єктів TreeNode. Так як об'єкт TreeNode містить властивість ChildNodes, яка може містити об'єкти TreeNode, то клас TreeNodeCollection є ієрархічною структурою даних, що представляє всі вузли елементу управління TreeView.

Типи вузлів TreeView

Елемент управління TreeView складається з одного або декількох вузлів. *Вузлом* називається кожен запис у дереві, який представлений об'єктом TreeNode. У наведеній нижче таблиці описано три різних типи вузлів.

Тип вузла	Опис
кореневий	вузол без батьківського вузла і з одним або декількома дочірніми вузлами
батьківський	вузол з батьківським вузлом і з одним або декількома дочірніми вузлами
кінцевий	вузол без дочірніх вузлів

Незважаючи на те, що одне дерево має, як правило, тільки один кореневий вузол, елемент управління TreeView дозволяє додавання декількох елементів управління в структуру дерева. Подібні можливості корисні, коли є необхідність у відображенні списку без єдиного кореневого вузла, як, наприклад, в списку категорій продуктів.

Кожен вузол містить властивості Text і Value. Значення властивості Text відображається в елементі управління TreeView, а властивість Value використовується для зберігання додаткових даних про вузол, наприклад, даних, що передаються події зворотної передачі, яка пов'язана з вузлом.

Натискання на вузол елемента управління TreeView може або ініціювати подію вибору (через зворотню відправку), або перенаправити браузер на іншу сторінку. Якщо властивість NavigateUrl не задано, то при натисканні на вузол викликається подія SelectedNodeChanged, яку можна обробити для реалізації необхідної функціональності. Кожен вузол має властивість SelectAction, яку можна використовувати для визначення окремих дій, що відбуваються при натисканні на вузол, наприклад, можливість розгортати та згортати вузла. Щоб, при натисканні на вузол, замість ініціювання події вибору відбувалося перенаправлення на певну сторінку, потрібно присвоїти властивості NavigateUrl вузла значення, відмінне від порожнього рядка.

Заповнення даних TreeNode на вимогу

Іноді не дуже зручно визначати структуру даних статично, так як дані можуть залежати від інформації, що одержується під час виконання. Для динамічного відображення даних елемент управління TreeView підтримує динамічне заповнення вузла. Якщо елемент керування TreeView налаштований на заповнення на вимогу, то елемент управління ініціює подію, коли користувач розгортає вузол. Оброблювач подій витягує відповідні дані та заповнює вузол, на який натиснув користувач. Щоб заповнити об'єкт TreeNode даними на вимогу, потрібно присвоїти властивості PopulateOnDemand вузла значення true і створіть обробник подій TreeNodePopulate для заповнення даними об'єкта TreeNode.

Заповнення вузла в TreeView на стороні клієнта

Будь-який браузер, що має властивість SupportsCallback з встановленим значенням true у файлі конфігурації можливостей браузера, підтримує заповнення вузла на стороні клієнта.

Заповнення вузла на стороні клієнта дозволяє елементу управління TreeView заповнити вузол за допомогою виклику події TreeNodePopulate сервера з клієнтського сценарію, замість запиту обміну даними з сервером.

Дозвіл сценаріїв клієнта

За замовчуванням в сучасних браузерах розгортання та згортання вузлів в елементі управління TreeView виконуються на основі виклику клієнтських сценаріїв. Використання клієнтських сценаріїв зменшує інерційність взаємодії, так як елементу управління не треба обмінюватися даними з сервером для відображення нової інформації.

Зворотній передача в TreeView

За замовчуванням елемент управління TreeView управляє функціями розгортання та згортання на стороні клієнта за винятком тих випадків, коли браузер не підтримує клієнтські сценарії або властивість EnableClientScript має значення false. Якщо властивість PopulateNodesFromClient має значення true і браузер підтримує клієнтські сценарії, то елемент управління TreeView отримує дані з сервера, але не передає назад всю сторінку цілком.

Якщо елемент керування TreeView поставлений в режим вибору, то кожен раз при натисканні користувачем на вузол відбувається зворотна передача на сервер та викликається подія SelectedNodeChanged.

Зазвичай, управляти зворотною передачею подій варто тоді, коли елемент управління TreeView знаходиться в режимі вибору або вузли заповнюються динамічно. Це дієво, поки або властивість PopulateOnDemand, або властивість PopulateNodesFromClient мають значення true.

Використання елемента управління TreeView з елементами управління UpdatePanel

Елементи управління UpdatePanel використовуються для поновлення обраних областей сторінки замість оновлення всієї сторінки за допомогою зворотної передачі. Елемент управління TreeView можна використовувати всередині елемента керування UpdatePanel з такими обмеженнями:

– зворотні виклики елемента управління TreeView повинні бути пов’язані з асинхронними зворотними передачами, інакше перевірка події зворотного виклику завершиться невдачею. При присвоєнні властивості PopulateOnDemand елемента управління TreeNode значення true зворотні виклики дозволені. Щоб упевнитися, що зворотні виклики елемента управління TreeView працюють з елементом управління UpdatePanel, можна використовувати один з таких підходів:

– якщо елемент керування TreeView не всередині елемента керування UpdatePanel, то слід відключити зворотні виклики елементів управління TreeNode, які не є частиною асинхронної зворотної передачі. Для цього треба присвоїти властивості PopulateOnDemand значення false;

– оновити програмним шляхом всі елементи управління, які реєструють зворотні виклики під час асинхронної зворотної передачі. Наприклад, можна розмістити елемент управління TreeView всередині елемента керування UpdatePanel. Елемент управління TreeView не обов’язково повинен бути всередині елемента керування UpdatePanel, з якого відбуваються асинхронні зворотні передачі, так як елемент управління UpdatePanel, що містить елемент керування TreeView, оновлюється;

– слід застосовувати стилі шляхом посилання на клас таблиць каскадних стилів (CSS). Наприклад, замість того, щоб встановити атрибут властивості NodeStyle в формі «властивість.підлегла_властивість», задайте стиль за допомогою атрибута в формі «CssClass = ім’я_класу». Аналогічно, при використанні шаблону NodeStyle для завдання стилю, використовуйте атрибут CssClass шаблону.

– властивість EnableClientScript повинна мати значення true (значення за замовчуванням). Крім того, якщо зворотні виклики включені для елемента управління TreeView, то не можна змінювати властивість EnableClientScript між асинхронними зворотними передачами.

Завдання

Створити базу даних з оптимальним розміщенням таблиць на двох вузлах. Написати web-інтерфейс (портал) для роботи з цією базою. Портал повинен вміти коректно обробляти дані що вводяться, робити вибірку даних з таблиць, вставляти, видаляти та змінювати дані в таблицях, розташованих в різних базах даних, зберігаючи цілісність розподіленої бази даних.

При реалізації інтерфейсу використовувати технології для передачі даних на сторінку без її перезавантаження.*Перелік предметних областей*

Студентська група	Міста України	Комп’ютери
Результати екзамену	Розклад занять	Товари
Відділення коледжу	Погода на тиждень	Автомобілі