# ADO.Net

1. Dataset

* Là một đối tượng có thể chứa nhiều Datatable cùng với mối liên hệ chữa chúng và kể cả các ràng buôc, được lưu hoàn toàn trong bộ nhớ để làm việc ofline.
* **Cấu trúc của DataSet và DataTable**
* **Namespace**: System.Data
* Một DataSet tương tự như một tập tin database vật lý hoàn chỉnh nhưng được lưu trong bộ nhớ. DataSet bao gồm các DataTable, DataTable bao gồm các DataColumn, DataRow, các constraint được minh họa như hình dưới:
* **DataSet**:

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| DataTableCollection | Tables | Gets the collection of tables contained in the System.Data.DataSet |
| DataRelationCollection | Relations | Get the collection of relations that link tables and allow navigation from parent tables to child tables. |

* **DataTable**:

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| DataColumnCollection | Columns | Gets the collection of columns that belong to this table. |
| DataRowCollection | Rows | Gets the collection of rows that belong to this table. |
| ConstraintCollection | Constraints | Gets the collection of constraints maintained by this table. |

1. **Các object .net cung cấp**

* **The Command Object**: Sử dụng the connection object để thực thi các truy vấn SQL
* Các truy vaanscos thể ở dạng form văn bản, thủ tục hoặc truy cập bảng trực tiếp
* Tính năng quan trọng của Command là có thể sử dụng để thực thi các truy vấn và thủ tục được lưu trữ với các tham số
* Những phương thức thường xuyên sử dụng trong Command Object:

+ ExecuteNoneQuery

+ ExecuteReader

+ ExecuteScalar

+ ExcuteXmlReader

* **The Connection** nằm giữa nguồn dữ liệu và Data Adapter( thông qua Command)
* Connection cũng có thể được kết nối với một đối tượng Command để thực hiện các truy vấn SQL, có thể được sử dụng để truy xuất, thêm, cập nhật và xóa dữ liệu vào nguồn dữ liệu.
* Connection có vai trò hữu ích trong việc tạo một transaction. Các giao dịch được lưu trữ trong transaction object và có tất cả các tính năng tố để xử lý các giao dịch như commit và rollback

1. Các tạo datagrid

* Gồm các thuộc tính cơ bản sau

+ Thông thường:

* Show header: Qui định dòng tiêu đề trên có được phép hiển thị hay không. (mặc định là có hiển thị dòng tiêu đề)
* Show footer: Qui định dòng tiêu đề dưới có được phép hiển thị hay không. (mặc định là không hiển thị dòng tiêu đề dưới)
* Allow sorting: Có cho phép sắp xếp dữ liệu hay không. (mặc định là không cho phép sắp xếp)

+ Columns

* Create columns automatically at runtime: Khi chọn chức năng này, DataGrid sẽ tự động phát sinh đầy đủ các cột có trong nguồn dữ liệu. Nếu chúng ta muốn qui định các cột cần hiển thị, chúng ta không chọn chức năng này.
* Column list: Qui định các cột được hiển thị trong lưới.
* Bound Column: Cột có liên kết với nguồn dữ liệu.
* Button Column: Cột dạng nút lệnh đã được thiết kế sẵn. Điều khiển DataGrid cung cấp cho chúng ta 3 loại cột dạng này:
* Select: Nút lệnh chọn dòng dữ liệu
* Edit, Cancel, Update: Các nút lệnh hỗ trợ chức năng cập nhật dữ liệu trực tiếp trên lưới.
* Delete: Nút lệnh xóa dòng dữ liệu
* Template Column: Cột do người dùng tự thiết kế

1. Tìm hiểu Datareader – Dataview

* DataReader là một trình lặp chỉ chuyển tiếp trên một tập hợp các kết quả. Đây thường là cách hiệu quả nhất để xử lý các bản ghi khi bạn không cần quyền truy cập ngẫu nhiên (nói cách khác là bạn không thể quay ngược lại). Nó "có thể mở rộng" cho bất kỳ số lượng bản ghi, ít nhất là về áp lực bộ nhớ, vì nó chỉ tải một bản ghi tại một thời điểm. Một cách điển hình để lấy DataReader là sử dụng phương thức ExecuteReader của DbCommand
* DataView giống như một tập con ảo của DataTable..

1. Datalist

Thiết kế hình thức hiển thị cho DataList cũng tương tự như thiết kế cho cột Template Column của DataGrid.

1. Binding Data

- Kỹ thuật liên kết một nguồn dữ liệu với nơi sử dụng dữ liệu, và đồng bộ hóa dữ liệu giữa chúng

Cú pháp Data Binding:

* Việc gắn dữ liệu vào trang ASP.NET trực tiếp với mã HTML được đánh dấu bắt đầu bằng ký hiệu <%# và kết thúc bằng ký hiệu %>
* Ở phần code behind, chúng ta phải sử dụng Phương thức DataBind, nếu không sever sẽ hiểu đoạn mã trên như một đoạn html bình thường và bỏ qua

+ Binding với các thuộc tính (Property)

Vd: <strong><em>Tên trình duyệt: <%# Request.Browser.Browser %> <br /> Phiên bản: <%# Request.Browser.MajorVersion %> <br /> Hệ điều hành: <%# Request.Browser.Platform %> <br /></em></strong>

+ Gắn Binding với thuộc tính của các Control

Vd: <asp:TextBox runat=”server” id=”txtName” Text=”<%# MyName %>” />

1. Đọc data trong file xml

* Trong một số trường hợp, việc lưu trữ giữ liệu vào các file XML là giải pháp tốt nhất, chúng thường được sử dụng để lưu trữ các thiết đặt hiển thị hoặc một số thông tin không cần bảo mật. Tương tự như với file text, tuy nhiên nó được tổ chức theo một cấu trúc dữ liệu và dễ dàng truy vấn sử lý hơn file thông thường.
* Chúng ta sử dụng XmlDocument để đọc XML từ chuỗi hoặc từ tệp.