بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

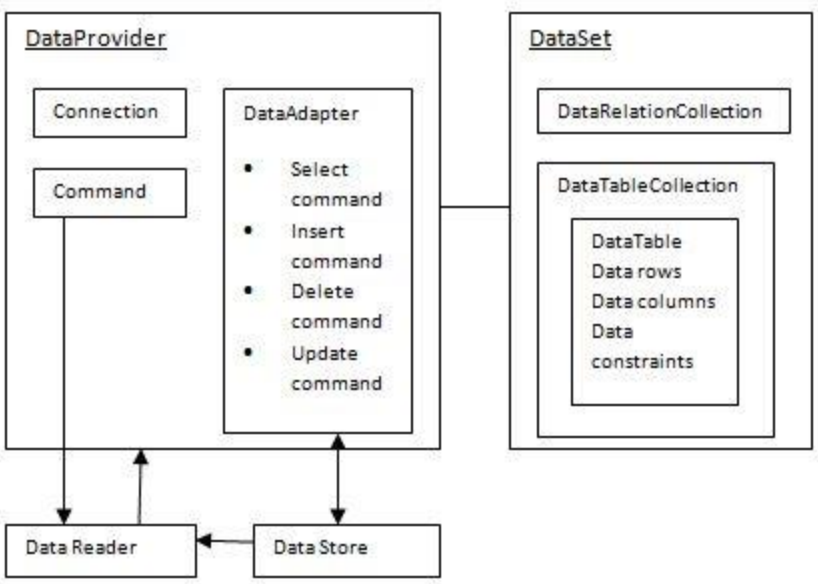
آموزش ADO.NET و کاربرد آن درASP

مدرس : مهندس افشین رفوآ

آموزش ADO.NET و کاربرد آن در ASP

ADO.NET یک پل ارتباطی بین کنترل های front end و پایگاه داده ی back end فراهم می آورد. اشیا ADO.NET تمامی عملیات دسترسی به داده را کپسوله سازی می کند. کنترل ها برای نمایش داده با این اشیا تعامل و ارتباط برقرار می کند، به همین دلیل جزئیات جا به جایی داده ها پنهان می شود.

شکل زیر اشیا ADO.NET را به صورت مختصر نمایش می دهد.



کلاس DataSet

مجموعه داده ای (data set) در واقع زیر مجموعه ای از پایگاه داده (data base) محسوب می شود. این کلاس یک ارتباط پیوسته با پایگاه داده ندارد. جهت بروز رسانی پایگاه داده به اتصال دوباره (reconnection) احتیاج است. کلاس DataSet دربردارنده ی اشیا DataTable و DataRelation می باشد. اشیا DataRelation بیانگر ارتباط بین دو جدول می باشد.

جدول زیر برخی از خاصیت های (property) مهم و کارامد این کلاس را فهرست می کند.

شرح	خواص
این خاصیت مشخص می کند آیا مقایسه ی رشته ها داخل جداول داده به کوچک و بزرگی حروف حساس هستند یا خیر.	CaseSensitive
ظرف دربردارنده ی مولفه ی (component) مورد نظر را برمی گرداند.	Container
اسم data set (مجموعه داده) جاری را گرفته یا تنظیم می کند.	DataSetName
به وسیله ی این خاصیت می توان نمایی (view) از داده های موجود در data set برگرداند که امکان فیلتر کردن، جستجو و پیمایش را با استفاده از یک DataViewManager سفارشی فراهم می آورد.	DefaultViewManager
این خاصیت نشان می دهد آیا مولفه (کامپوننت) مورد نظر هم اکنون در حالت design mode قرار دارد یا خیر.	DesignMode
مشخص می کند آیا به هنگام اجرای هرگونه عملیات بروز رسانی، قوانینی که محدودیت اعمال می کنند (constraint rule) لحاظ شده و رعایت می شوند یا خیر.	EnforceConstraints
فهرستی از event handler های متصل به کامپوننت مورد نظر را برمی گرداند.	Events

ExtendedProperties	مجموعه اطلاعات سفارشی یا اختصاصی کاربر که مربوط به کلاس DataSet می باشد را باز می گرداند.
HasErrors	این خاصیت مشخص می کند آیا خطا یا اشکالی وجود دارد یا خیر.
IsInitialized	نشان می دهد آیا DataSet مقداردهی اولیه شده است یا خیر.
Locale	اطلاعات محلی که به منظور مقایسه کردن رشته های داخل جدول بکار می رود را تنظیم (getیا set) می کند.
Namespace	این خاصیت فضای نام (namespace) کلاس DataSet را تنظیم (get یا set) می کند.
Prefix	خاصیت prefix به منظور شناسایی المان هایی که به کلاس DataSet تعلق دارند در سرتاسر یک سند XML بکار می رود.
Relations	آرایه یا مجموعه ای از اشیا کلاس DataRelation برمی گرداند.
Tables	آرایه ای (collection) از اشیا DataTable بازمی گرداند.

جدول زیر نیز تعدادی از متدهای مهم و پرکاربرد کلاس DataSet را ارائه می دهد.

توابع	شرح
AcceptChanges	این متد کلیه ی تغییرات یا اصلاحاتی که از زمان بارگذاری کلاس DataSet و یا فراخوانی تابع مزبور (AcceptChanges) اعمال شده را حفظ کرده و می پذیرد.
BeginInit	مقداردهی اولیه ی کلاس DataSet را راه اندازی می کند. فرایند مقداردهی اولیه درست در زمان اجرا (run time) رخ می دهد.
Clear	متد Clear کلیه ی داده ها را حذف می کند.

Clone	ساختار کلاس DataSet شامل شِما / DataTable schema ، رابطه ها (relations)، محدودیت ها (constraints) را کپی (نمونه سازی) می کند. توجه داشته باشید که این متد خود داده ها را کپی نمی کند.
Copy	این تابع هم ساختار و هم خود داده را کپی می کند.
CreateDataReader()	به ازای هر DataTable یک مجموعه از شی DataTableReader، به همان ترتیبی که جداول در مجموعه Tables (collection) قرار می گیرند (ظاهر می شوند)، برمی گرداند.
CreateDataReader(DataTable[])	به ازای هر DataTable یک شی DataTableReader برمی گرداند.
EndInit	مقداردهی اولیه ی مجموعه داده را خاتمه می دهد.
Equals(Object)	تعیین می کند آیا شی مورد نظر با شی جاری برابر هست یا خیر.
Finalize	منابع را آزاده ساخته و امکان اجرای دیگر عملیات پاک سازی را فراهم می آورد.
GetChanges	یک کپی از DataSet به همراه تمامی تغییراتی که از زمان بارگذاری این کلاس یا فراخوانی متد AcceptChanges اعمال شده، بازمی گرداند.
GetChanges(DataRowState)	یک کپی از DataSet به همراه تمامی تغییراتی که از زمان بارگذاری این کلاس یا فراخوانی متد AcceptChanges ایجاد شده برمی گرداند که توسط شئ DataRowState فیلتر می شود.
GetDataSetSchema	کپی از XmlSchemaSet برای DataSet برمی گرداند.
GetObjectData	اطلاعات لازم برای سریال کردن DataSet را بازگردانی می نماید.
GetType	نوع نمونه (instance) جاری را بر می گرداند.
GetXML	داده را در قالب XML برمی گرداند.

GetXMLSchema	شِمای XSD را برای نمایش داده ها در قالب XML برمی گرداند.
HasChanges()	این تابع مقداری برمی گرداند که نشانگر وجود یا عدم وجود تغییر و اصلاحات در DataSet می باشد (این تغییرات شامل سطرهای (row) جدید، پاک یا اصلاح شده می باشد).
HasChanges(DataRowState)	این تابع مقداری برمی گرداند که نشانگر وجود یا عدم وجود تغییر و اصلاحات در DataSet می باشد (این تغییرات شامل سطرهای (row)جدید، پاک یا اصلاح شده می باشد) که توسط شئ DataRowState فیلتر می شود.
IsBinarySerialized	قالب (فرمت) نمایش DataSet سریال شده را بررسی می کند.
Load(IDataReader, LoadOption, DataTable[])	DataSet را با مقادیر ی از یک منبع داده ای با استفاده از IDataReader ارائه شده و آرایه ای از نمونه های DataTable پر می کند تا از این طریق اطلاعات شِما و فضای نام را فراهم کند.
Load(IDataReader, LoadOption, String[])	DataSet را با مقادیر ی از یک منبع داده ای با استفاده از IDataReader ارائه شده و آرایه ای از رشته ها پر می کند تا بدین وسیله اسامی لازم برای جداول درون DataSet فراهم شود.
Merge()	داده ها را با داده هایی مشتق شده از یک DataSet دیگر ادغام می کند. این متد دارای اشکال overload شده ی مختلفی می باشد.
ReadXML()	داده ها و شمای XML را از درون DataSet می خواند. این متد فرم های overload شده ی مختلفی دارد.
ReadXMLSchema (0)	یک شمای XML را از DataSet می خواند. این تابع حالت های overload شده ی مختلفی دارد.
RejectChanges	تمامی تغییرات اعمال شده از زمانی که متد AcceptChanges برای آخرین بار فراخوانده شد را به حالت اول برمی گرداند.
WriteXML()	داده های موجود را بر اساس DataSet به صورت XML می

نویسد. این تابع به صورت های مختلف overload شده است.	
ساختار DataSet را به صورت XML (در قالب یا با شِمای XML) می نویسد. این متد به اشکال مختف overload می شود.	WriteXMLSchema ()

کلاس DataTable

این کلاس همان طور که از اسم آن پیدا است نمایشگر جداول داخل پایگاه داده است. کلاس گفته شده دارای خواص (property) زیر می باشد که بیشتر آن ها به استثنای خاصیت PrimaryKey فقط قابل خواندن (read-only) هستند.

شرح	خاصیت
مجموعه رابط های فرزندی (child relation) را بازمی گرداند.	ChildRelations
مجموعه ستون هایی که متعلق به جدول مورد نظر می باشند را برمی گرداند.	Columns
مجموعه محدودیت هایی (constraint) که در این جدول (در رابطه با بروز رسانی یا حذف مقدار در ستون های جدول) اعمال می شوند را می گیرد.	Constraints
DataSet والد را برمی گرداند.	DataSet
این خاصیت یک DataView برمی گرداند که می توان با استفاده از آن یک DataTable را مرتب ساخت، فیلتر کرد یا در آن جستجو (search) انجام داد. به عبارتی دیگر نما یا view سفارشی از جدول مورد نظر که می تواند شامل نمای فیلتر شده یا موقعیت مکان نمای موس باشد برمی گرداند.	DefaultView
مجموعه (collection) رابط های پدری (parent relation) را برای این DataTable بازمی گرداند.	ParentRelations
آرایه ای از ستون ها بر گردانده یا تنظیم می کند که به مثابه ی کلید	PrimaryKey

اولیه برای جدول مورد نظر عمل می کند (بکار گرفته می شود).	
مجموعه سطرهای متعلق به این جدول را بازمی گرداند.	Rows

جدول زیر متدهای پرکاربرد کلاس DataTable را برمی گرداند.

شرح	تابع
کلیه ی تغییراتی که از زمان آخرین فراخوانی متد AcceptChanges به این جدول وارد شده را پذیرفته و اعمال می کند.	AcceptChanges
این تابع تمامی داده ها را از جدول حذف می کند.	Clear
نسخه ای (کپی) از DataTable به همراه تمامی تغییرات وارد آمده از زمان فراخوانی متد AcceptChanges را برمی گرداند.	GetChanges
آرایه ای از سطرهایی که دارای خطا هستند را برمی گرداند.	GetErrors
یک سطر جدید را کپی کرده و در جدول مورد نظر جای گذاری می کند.	ImportRows
سطر مورد نظر را یافته، سپس آن را به روز رسانی می کند و در صورت نیافتن آن یک سطر جدید ایجاد می کند.	LoadDataRow
جدول جاری را با یک DataTable دیگر ادغام یا ترکیب می کند.	Merge
این تابع یک DataRow جدید (مطابق با شِمای جدول) ایجاد می کند.	NewRow
تمامی تغییراتی که از زمان آخرین فراخوانی متد AcceptChanges یا بار گذاری جدول مورد نظر به آن اعمال شده را به حالت اولیه برمی گرداند (تغییرات وارد آمده را رد می کند).	RejectChanges
این متد جدول مورد نظر را به حالت اولیه ی خود بازمی گرداند.	Reset
این تابع آرایه ای از اشیا DataRow برمی گرداند.	Select

کلاس DataRow

شی DataRow نشانگر سطر داخل جدول می باشد. خواص کلاس DataRow به ترتیب در جدول زیر فهرست شده.

شرح	خاصیت
مشخص می کند آیا در سطر مورد نظر خطا یا اشکالی وجود دارد یا خیر.	HasErrors
داده های ذخیره شده داخل ستون معین را گرفته یا تنظیم می کند.	Items
تمامی مقادیر سطر مورد نظر را از طریق آرایه گرفته یا تنظیم می کند.	ItemArrays
جدول پدر را بازمی گرداند.	Table

متدهای این کلاس به ترتیب زیر می باشند.

شرح	متد
کلیه ی تغییرات وارد آمده (به سطر مورد نظر) از زمان فراخوانی این متد را پذیرفته و به آن سطر اعمال می کند.	AcceptChanges
این تابع عملیات ویرایش را آغاز می کند.	BeginEdit
این متد عملیات ویراست را لغو می کند.	CancelEdit
DataRow را پاک می کند.	Delete
عملیات ویرایش را خاتمه می دهد.	EndEdit
سطرهای فرزند یا زیرمجموعه ی سطر مورد نظر را بازمی گرداند.	GetChildRows
سطر پدر را برمی گرداند.	GetParentRow
سطرهای والد شی DataRow را برمی گرداند.	GetParentRows
کلیه ی تغییراتی که از زمان آخرین فراخوانی تابع AcceptChanges به سطر مورد نظر اعمال شده را رد می کند (پس می زند).	RejectChanges

شی DataAdapter

شی نام برده به عنوان یک میانجی بین شی DataSet و پایگاه داده عمل می کند. DataAdapter به شی DataSet امکان می دهد به طور همزمان از چندین پایگاه داده یا دیگر منابع داده ای (data source) اطلاعات (داده) پذیرفته و در خود جای دهد.

شی DataReader

شی DataReader در واقع می تواند جایگزینی برای ترکیب دو شی بالا باشد. شی نام برده یک نوع دسترسی مبتنی بر اتصال (از طریق اتصال) به رکودهای داده (data records) موجود در پایگاه داده فراهمی می کند. این اشیا تنها ویژه ی دسترسی read-only (دسترسی که در آن تنها می توان داده را خواند و نه تغییر داد) تعبیه شده اند، مانند پر کردن لیست و قطع کردن اتصال به پایگاه داده.

شی های DbCommand و DbConnection

شی DbConnection نشانگر اتصال به منبع داده (data source) می باشد. اتصال (connection) می تواند بین command object های مختلف به اشتراک گذاشته شود.

شی DbCommand نشانگر دستور SQL یا رویه ی ذخیره شده (stored procedure) می باشد که به منظور بازیابی یا دستکاری و مدیریت داده به پایگاه داده فرستاده می شود.

مثال

در این مثال یک جدول ایجاد کرده و به داخل آن ستون، سطر، داده اضافه می کنیم، سپس با استفاده از شی GridView جدول را نمایش می دهیم.

Source file code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Ado_Example.aspx.cs"
Inherits="Ado_net.Ado_Example" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
      <title>
         Untitled Page
      </title>
   </head>
   <body>
      <form id="form1" runat="server">
         <div>
```

```
            <asp:GridView ID="GridView1" runat="server">
            </asp:GridView>
        </div>
    </form>
</body>

</html>
```

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        DataSet ds = CreateDataSet();
        GridView1.DataSource = ds.Tables["Student"];
        GridView1.DataBind();
    }
}
private DataSet CreateDataSet()
{
    //creating a DataSet object for tables
    DataSet dataset = new DataSet();

    // creating the student table
    DataTable Students = CreateStudentTable();
    dataset.Tables.Add(Students);
    return dataset;
}
private DataTable CreateStudentTable()
{
    DataTable Students = new DataTable("Student");

    // adding columns
    AddNewColumn(Students, "System.Int32", "StudentID");
    AddNewColumn(Students, "System.String", "StudentName");
    AddNewColumn(Students, "System.String", "StudentCity");
    // adding rows
    AddNewRow(Students, 1, "M H Kabir", "Kolkata");
    AddNewRow(Students, 1, "Shreya Sharma", "Delhi");
    AddNewRow(Students, 1, "Rini Mukherjee", "Hyderabad");
    AddNewRow(Students, 1, "Sunil Dubey", "Bikaner");
    AddNewRow(Students, 1, "Rajat Mishra", "Patna");
    return Students;
}
private void AddNewColumn(DataTable table, string columnType, string columnName)
{
    DataColumn column = table.Columns.Add(columnName, Type.GetType(columnType));
}
//adding data into the table
private void AddNewRow(DataTable table, int id, string name, string city)
{
    DataRow newrow = table.NewRow();
    newrow["StudentID"] = id;
    newrow["StudentName"] = name;
    newrow["StudentCity"] = city;
    table.Rows.Add(newrow);
}
```

پس از اجرای برنامه نتایج زیر را مشاهده می کنید.

اپلیکیشن ابتدا یک data set (مجموعه داده) ایجاد کرده، سپس با استفاده از تابع DataBind() (کنترل GridView) آن را به کنترل grid view متصل می کند.

تابع Createdataset() تابعی است که توسط کاربر تعریف (user-defined) شده. متد مزبور یک شی DataSet ایجاد کرده، سپس تابع CreateStudentTable() (که یک تابع user-defined دیگر می باشد) را فرامی خواند و از این طریق جدول مورد نیاز را ایجاد کرده و آن را به Tables collection مجموعه داده (data set) الحاق می کند.

متد CreateStudentTable() توابع ارائه شده توسط کاربر AddNewColumn() و AddNewRow() را فرا می خواند. دو تابع بیان شده ستون ها و سطرهای جدول مورد نظر را ساخته و داده های لازم را درون سطرها جای گذاری می کنند.

پس از اجرای صفحه، سطرهای جدول به صورت زیر برگردانده می شوند.