**ADO.NET offers two data access modes**  
**1.** Connection oriented data access  
**2.** Disconnected data access  
  
In this video, we will discuss **disconnected data access.**SqlDataAdapter and DataSet objects together provide disconnected data access.  
  
   
  
A DataSet is an in-memory data store that can hold one or more tables. DataSets only hold data and do not interact with the underlying database table. The DataSet object has no knowledge of the underlying Data Source. It is the SqlDataAdapter object that retrieves data from the datasource.   
  
**This is how it works.**  
**1.** You create an instance of SqlDataAdapter by specifying a select command and a connection object  
string connectionString = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;  
SqlConnection connection = new SqlConnection(connectionString);  
string selectQuery = "Select \* from tblStudents";  
SqlDataAdapter dataAdapter = new SqlDataAdapter(selectQuery, connection);  
  
**2.** When **SqlDataAdapter.Fill()** method is invoked, SqlDataAdapter opens the connection to the database, executes the select command, and the DataSet is populated with the data that is retrieved. The SqlDataAdapter automatically closes the connection.  
DataSet dataSet = new DataSet();  
dataAdapter.Fill(dataSet, "Students");  
  
**3.** You now have data in the DataSet and there is no active connection to the database. At this point you can make any changes(insert, update, delete) to the data in the DataSet. Only the data in the DataSet is changed, the underlying database table data is not changed.  
  
**4.** To update the underlying database table, invoke SqlDataAdapter.Update() method. Make sure there is an UPDATE, DELETE and INSERT command are associated with SqlDataAdapter object when Update() method is called, otherwise there would be a runtime exception.  
dataAdapter.Update(DataSetObject, "Students");  
  
**ASPX Code:**  
<div style="font-family: Arial">  
    <asp:Button ID="btnGetDataFromDB" runat="server" Text="Get Data from Database"   
        onclick="btnGetDataFromDB\_Click" />  
    <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"   
        DataKeyNames="ID" onrowediting="GridView1\_RowEditing"   
        onrowcancelingedit="GridView1\_RowCancelingEdit"   
        onrowdeleting="GridView1\_RowDeleting"   
        onrowupdating="GridView1\_RowUpdating">  
        <Columns>  
            <asp:CommandField ShowDeleteButton="True" ShowEditButton="True" />  
            <asp:BoundField DataField="ID" HeaderText="ID" InsertVisible="False"   
                ReadOnly="True" SortExpression="ID" />  
            <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />  
            <asp:BoundField DataField="Gender" HeaderText="Gender"   
                SortExpression="Gender" />  
            <asp:BoundField DataField="TotalMarks" HeaderText="TotalMarks"   
                SortExpression="TotalMarks" />  
        </Columns>  
    </asp:GridView>  
    <asp:Button ID="btnUpdateDatabaseTable" runat="server"   
        Text="Update Database Table" onclick="btnUpdateDatabaseTable\_Click" />  
    <asp:Label ID="lblStatus" runat="server"></asp:Label>  
</div>  
  
**ASPX.CS Code:**  
public partial class WebForm1 : System.Web.UI.Page  
{  
    protected void Page\_Load(object sender, EventArgs e)  
    {  
    }  
  
    private void GetDataFromDB()  
    {  
        string connectionString = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;  
        SqlConnection connection = new SqlConnection(connectionString);  
        string selectQuery = "Select \* from tblStudents";  
        SqlDataAdapter dataAdapter = new SqlDataAdapter(selectQuery, connection);  
  
        DataSet dataSet = new DataSet();  
        dataAdapter.Fill(dataSet, "Students");  
        // Set ID column as the primary key  
        dataSet.Tables["Students"].PrimaryKey =  
            new DataColumn[] { dataSet.Tables["Students"].Columns["ID"] };  
        // Store the dataset in Cache  
        Cache.Insert("DATASET", dataSet, null, DateTime.Now.AddHours(24),  
            System.Web.Caching.Cache.NoSlidingExpiration);  
  
        GridView1.DataSource = dataSet;  
        GridView1.DataBind();  
  
        lblStatus.Text = "Data loded from Database";  
    }  
  
    private void GetDataFromCache()  
    {  
        if (Cache["DATASET"] != null)  
        {  
            GridView1.DataSource = (DataSet)Cache["DATASET"];  
            GridView1.DataBind();  
        }  
    }  
  
    protected void GridView1\_RowEditing(object sender, GridViewEditEventArgs e)  
    {  
        // Set row in editing mode  
        GridView1.EditIndex = e.NewEditIndex;  
        GetDataFromCache();  
    }  
  
    protected void GridView1\_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)  
    {  
        GridView1.EditIndex = -1;  
        GetDataFromCache();  
    }  
  
    protected void GridView1\_RowUpdating(object sender, GridViewUpdateEventArgs e)  
    {  
        // Retrieve dataset from cache  
        DataSet dataSet = (DataSet)Cache["DATASET"];  
        // Find datarow to edit using primay key  
        DataRow dataRow = dataSet.Tables["Students"].Rows.Find(e.Keys["ID"]);  
        // Update datarow values  
        dataRow["Name"] = e.NewValues["Name"];  
        dataRow["Gender"] = e.NewValues["Gender"];  
        dataRow["TotalMarks"] = e.NewValues["TotalMarks"];  
        // Overwrite the dataset in cache  
        Cache.Insert("DATASET", dataSet, null, DateTime.Now.AddHours(24),  
            System.Web.Caching.Cache.NoSlidingExpiration);  
        // Remove the row from edit mode  
        GridView1.EditIndex = -1;  
        // Reload data to gridview from cache  
        GetDataFromCache();  
    }  
  
    protected void GridView1\_RowDeleting(object sender, GridViewDeleteEventArgs e)  
    {  
        DataSet dataSet = (DataSet)Cache["DATASET"];  
        dataSet.Tables["Students"].Rows.Find(e.Keys["ID"]).Delete();  
        Cache.Insert("DATASET", dataSet, null, DateTime.Now.AddHours(24),  
            System.Web.Caching.Cache.NoSlidingExpiration);  
        GetDataFromCache();  
    }  
  
    protected void btnGetDataFromDB\_Click(object sender, EventArgs e)  
    {  
        GetDataFromDB();  
    }  
  
    protected void btnUpdateDatabaseTable\_Click(object sender, EventArgs e)  
    {  
        if (Cache["DATASET"] != null)  
        {  
            string connectionString =  
            ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;  
            SqlConnection connection = new SqlConnection(connectionString);  
            string selectQuery = "Select \* from tblStudents";  
            SqlDataAdapter dataAdapter = new SqlDataAdapter(selectQuery, connection);  
  
            // Update command to update database table  
            string strUpdateCommand = "Update tblStudents set Name = @Name, Gender = @Gender, TotalMarks = @TotalMarks where Id = @Id";  
            // Create an instance of SqlCommand using the update command created above  
            SqlCommand updateCommand = new SqlCommand(strUpdateCommand, connection);  
            // Specify the parameters of the update command  
            updateCommand.Parameters.Add("@Name", SqlDbType.NVarChar, 50, "Name");  
            updateCommand.Parameters.Add("@Gender", SqlDbType.NVarChar, 20, "Gender");  
            updateCommand.Parameters.Add("@TotalMarks", SqlDbType.Int, 0, "TotalMarks");  
            updateCommand.Parameters.Add("@Id", SqlDbType.Int, 0, "Id");  
            // Associate update command with SqlDataAdapter instance  
            dataAdapter.UpdateCommand = updateCommand;  
  
            // Delete command to delete data from database table  
            string strDeleteCommand = "Delete from tblStudents where Id = @Id";  
            // Create an instance of SqlCommand using the delete command created above  
            SqlCommand deleteCommand = new SqlCommand(strDeleteCommand, connection);  
            // Specify the parameters of the delete command  
            deleteCommand.Parameters.Add("@Id", SqlDbType.Int, 0, "Id");  
            // Associate delete command with SqlDataAdapter instance  
            dataAdapter.DeleteCommand = deleteCommand;  
  
            // Update the underlying database table  
            dataAdapter.Update((DataSet)Cache["DATASET"], "Students");  
            lblStatus.Text = "Database table updated";  
        }  
    }  
}  
  
**Please make sure to include the following using declarations:**  
using System.Configuration;  
using System.Data;  
using System.Data.SqlClient;