

إضافة CheckBox لاختيار الأسطر في ال GridView

الكاتب: فراس دركزلي

الإصدار: 1.0

كتبت في: ٢٠٠٩-٢-٢٧

سنشرح في هذه المقالة كيفية إضافة CheckBox إلى أسطر ال GridView من أجل السماح للمستخدم باختيار مجموعة من الأسطر لتنفيذ إجراء محدد على هذه الأسطر (حذف هذه الأسطر مثلاً).

- بناء مثال للاعتماد عليه في هذه المقالة

سنبدأ أولاً بعمل GridView و ملأها ببعض البيانات الوهمية (Dummy Data) من أجل تجربة محتوى هذه المقالة عليها. ستكون خطوات العمل كالتالي:

- ١- عمل مشروع جديد ASP.NET Web Application
- ٢- كتابة صف (class) لتمثيل الطالب من خلال مجموعة من الخصائص.
- ٣- تعريف مصفوفة من الطلاب و ملؤها بالبيانات.
- ٤- إضافة GridView و تحديد الاعمدة التي نريد عرضها من خلالها.
- ٥- إسناد قيم عناصر المصفوفة إلى ال GridView.
- ٦- إضافة عمود من أجل السماح للمستخدم باختيار الأسطر.
- ٧- الحصول على الأسطر التي اختارها المستخدم.
- ٨- إضافة CheckBox لاختيار جميع الأسطر دفعةً واحدة.

لنبدأ الآن بعمل هذه الخطوات:

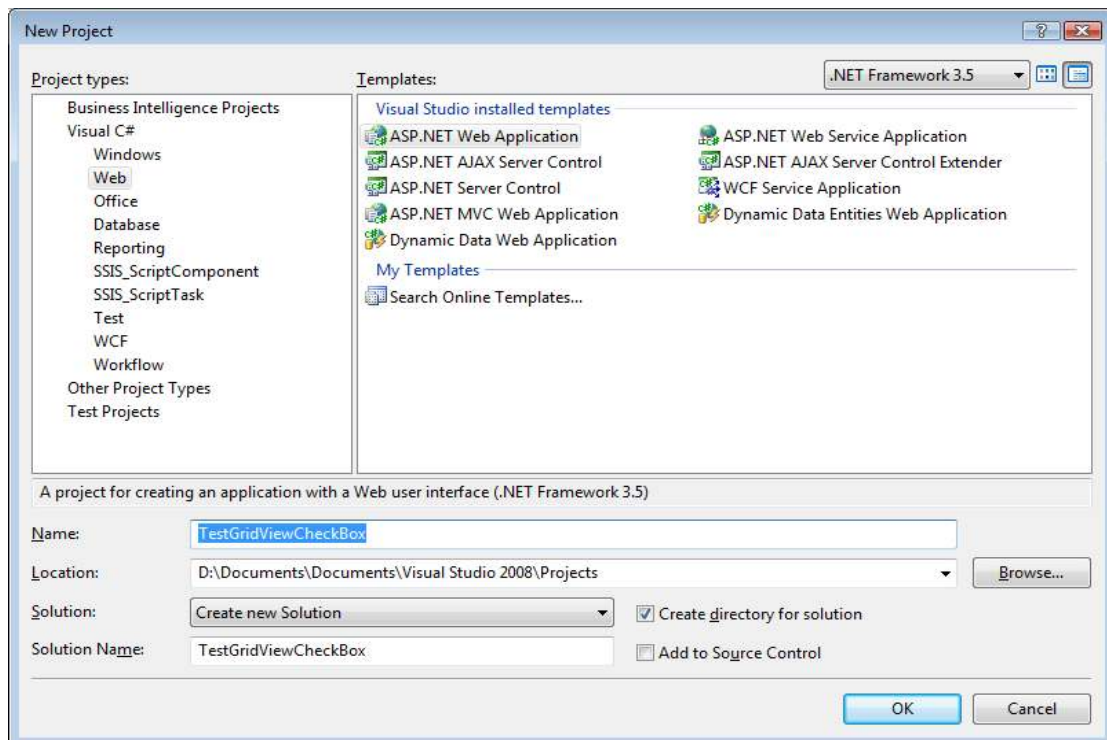
١- عمل مشروع جديد من نوع ASP.NET Web Application

سنعتمد في هذا المثال على استخدام Visual Studio 2008 للدعم التلقائي للمشاريع من النوع ASP.NET Web Application

ملاحظة:

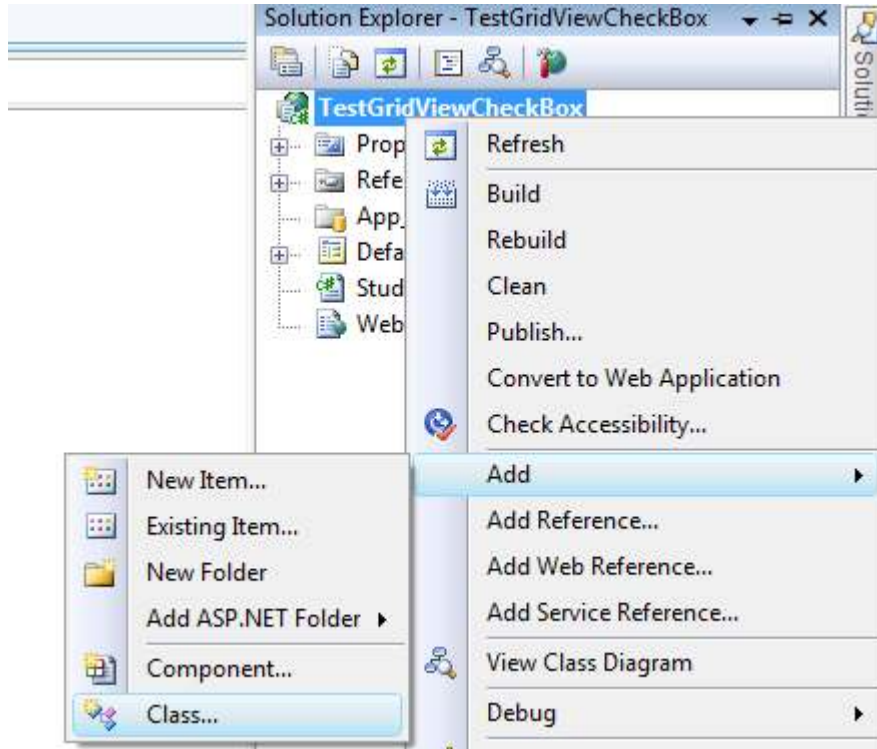
- في حال عدم توفر Visual Studio 2008، يمكن استخدام المشروع من النوع ASP.NET Web Site في Visual Studio 2005 كبديل له.

لنقم أولاً بعمل مشروع جديد باسم TestGridViewCheckBox كما توضح الصورة التالية:



٢- إضافة النمط الخاص بالطالب:

سنقوم بإضافة عنصر جديد من نوع class إلى المشروع و ذلك بالضغط على اسم المشروع في الـ Solution Explorer بالزر اليميني للفأرة و اختيار Add و بعدها Class.



ستظهر لنا نافذة نقوم من خلالها بكتابة اسم الصف المطلوب (Student.cs)، حيث سيتم فتح النافذة الخاصة بهذا الـ Class لكتابة الكود الخاص به.

سنقوم بتعريف ثلاثة خصائص هي:

- رقم الطالب
- اسم الطالب
- معدل الطالب

سيكون الكود الخاص بالصف Student كالتالي:

```
public class Student
{
    private int id;

    public int Id
    {
        set { this.id = value; }
        get { return this.id; }
    }

    private string name;

    public string Name
    {
        set { this.name = value; }
        get { return this.name; }
    }
}
```

```
private short average;

public short Average
{
    set { this.average = value; }
    get { return this.average; }
}
}
```

ملاحظة:

- يمكن الاستفادة من إحدى المزايا الجديدة في C# 3.0 من خلال Visual Studio 2008 وهي Auto Implemented Properties، و عندها سيكون الكود السابق كما يلي:

```
public class Student
{
    public int Id { get; set; }

    public string Name { get; set; }

    public short Average { get; set; }
}
```

ملاحظة:

- يمكن مراجعة الرابط التالي لشرح عن الـ Auto Implemented Properties <http://dotnet.org.za/hiltong/archive/2008/01/20/auto-implemented-properties-amp-object-initializers.aspx>

سنقوم بكتابة باني غير افتراضي (Parameterized Constructor) لتسهيل عملية تعريف العناصر من النمط Student كما يلي:

```
public Student(int id, string name, short average)
{
    this.Id = id;
    this.Name = name;
    this.Average = average;
}
```

٣- تعريف المصفوفة الخاصة بالطلاب:

سنقوم الآن بكتابة Static Method تعيد لنا مصفوفة عناصر من النمط Student (لتعبئة قيمها في الـ GridView) كما يلي:

```
public static List<Student> GetSampleList()
{
    Student s1 = new Student(1, "Feras", 64);
    Student s2 = new Student(2, "Samer", 93);
    Student s3 = new Student(3, "Yamen", 72);

    return new List<Student>() { s1, s2, s3 };
}
```

سيكون الآن النمط Student كما يلي:

```
public class Student
{
    public int Id { get; set; }

    public string Name { get; set; }

    public short Average { get; set; }
}
```

```

public Student(int id, string name, short average)
{
    this.Id = id;
    this.Name = name;
    this.Average = average;
}

public static List<Student> GetSampleList()
{
    Student s1 = new Student(1, "Feras", 64);
    Student s2 = new Student(2, "Samer", 93);
    Student s3 = new Student(3, "Yamen", 72);

    return new List<Student>() { s1, s2, s3 };
}
}

```

٤- إضافة الـ GridView إلى صفحة الـ HTML:

سنقوم الآن بكتابة الشيفرة الخاصة بإضافة الـ GridView مع تحديد الأعمدة الموجودة فيها كما يلي:

```

<asp:GridView runat="server" ID="gvTestData" AutoGenerateColumns="false">
    <Columns>
        <asp:BoundField DataField="Id" HeaderText="الطالب رقم">
            <ItemStyle Width="60px" HorizontalAlign="Center" />
        </asp:BoundField>
        <asp:BoundField DataField="Name" HeaderText="الطالب اسم">
            <ItemStyle Width="160px" HorizontalAlign="Center" />
        </asp:BoundField>
        <asp:BoundField DataField="Average" HeaderText="معدل">
            <ItemStyle Width="60px" HorizontalAlign="Center" />
        </asp:BoundField>
    </Columns>
</asp:GridView>

```

٥- إسناد مصفوفة الطلاب السابقة إلى الـ GridView:

إذا قمنا بتشغيل الصفحة السابقة (التي وضعنا فيها الـ GridView و التي هي في مثالنا هذا Default.aspx) فلن تظهر لنا أي نتيجة على الصفحة لأنه ليس هناك أية مصادر بيانات (Data Sources) قد تم إسنادها (Binding) إلى الـ GridView، و بالتالي فإن الـ GridView لن تعرض أية بيانات و لن تظهر أبداً ضمن الصفحة الموجودة لدينا.

سنقوم الآن بإسناد عناصر مصفوفة الطلاب إلى الـ GridView كي تقوم بعرض النتائج على الشاشة من خلال كتابة التعليمات التالية في الحدث Page_Load للصفحة المطلوبة:

```

gvTestData.DataSource = Student.GetSampleList();
gvTestData.DataBind();

```

تقوم التعليمات السابقة بإسناد ناتج الـ Method التي كتبناها في الصف Student إلى الـ GridView من خلال الخاصية DataSource، تقوم التعليمة DataBind بعملية نسخ البيانات فعلياً (من المصدر الذي حددناه في التعليمة السابقة) إلى الـ GridView.

لو قمنا الآن بتشغيل الصفحة الحالية ستظهر لنا كما في الصورة:

رقم الطالب	اسم الطالب	معدل
١	Feras	٦٤
٢	Samer	٩٣
٣	Yamen	٧٢

ملاحظة:

تتميز الـ GridView بأنها تقوم تلقائياً بحفظ البيانات ضمن الـ ViewState الخاصة بالصفحة، وبالتالي ليس هناك حاجة لإجراء عملية الـ Binding في كل مرة عند حدوث PostBack، لأن المعلومات موجودة في الـ GridView عند تحميل الصفحة أول مرة، و يتم حفظها مع الـ ViewState الخاصة بالصفحة، و عند عمل PostBack للصفحة يتم استخراج البيانات مرة أخرى من الـ ViewState و بالتالي ليس هناك حاجة لإحضار البيانات من مصدر البيانات مرة أخرى، لذلك سنقوم بإسناد البيانات إلى الـ GridView عند أول تحميل للصفحة فقط، و بالتالي سيكون الكود كالتالي:

```
if (!this.IsPostBack)
{
    gvTestData.DataSource = Student.GetSampleList();
    gvTestData.DataBind();
}
```

٦- إضافة عمود من أجل السماح للمستخدم باختيار الأسطر:

سنقوم الآن بإضافة عمود جديد إلى الـ GridView بحيث يظهر لنا CheckBox لكل سطر نستطيع من خلاله اختيار السطر المطلوب، و سنعمل ذلك كما يلي:

```
<asp:TemplateField>
    <ItemTemplate>
        <asp:CheckBox runat="server" ID="chkSelectRow" />
    </ItemTemplate>
    <ItemStyle Width="60px" HorizontalAlign="Center" />
</asp:TemplateField>
```

إذا قمنا بتشغيل الصفحة السابقة سيكون الشكل لدينا كالتالي:

رقم الطالب	اسم الطالب	معدل	
١	Feras	٦٤	☐
٢	Samer	٩٣	☒
٣	Yamen	٧٢	☒

حيث نستطيع الآن اختيار الأسطر المرغوبة من مجموع الأسطر الموجودة.

٧- الحصول على الأسطر التي اختارها المستخدم:

سنقوم الآن بإضافة زر (Button) إلى الصفحة من أجل عمل PostBack للصفحة للحصول على الأسطر التي قام المستخدم باختيارها و عرضها على الصفحة ضمن Label.

سنضيف أولاً الـ Button و الـ Label على الصفحة كما يلي:

```
<br />
<asp:Button runat="server" ID="btnGetSelectedStudents" Text="عرض"
    Width="80px" OnClick="btnGetSelectedStudents_Click" />
<br />
<br />
<asp:Label runat="server" ID="lblResult" />
```

و سنكتب في الحدث Click للزر btnGetSelectedStudents الكود التالي:

```
protected void btnGetSelectedStudents_Click(object sender, EventArgs e)
{
    StringBuilder builder = new StringBuilder();

    builder.Append("التالية الطلاب باختيار قمت لقد");
    builder.Append("<br />");
}
```

```

foreach (GridViewRow row in gvTestData.Rows)
{
    if (row.RowType == DataControlRowType.DataRow)
    {
        CheckBox c = (CheckBox)row.FindControl("chkSelectRow");
        string studentId = c.Attributes["StudentId"];

        if ((c != null) && (c.Checked == true))
        {
            builder.Append(studentId);
            builder.Append("<br />");
        }
    }
}

lblResult.Text = builder.ToString();
}

```

حيث يقوم هذه التعليمات بالمرور على أسطر الـ GridView (من النوع DataRow)، لأنه ليس لنا حاجة حالياً للتعامل مع الأسطر الممثلة للـ Header و الـ Footer في الـ GridView، ثم البحث عن العنصر **chkSelectRow** للحصول على حالته إذا كان Checked أم لا، و في حال تم اختياره سنقوم بإضافته إلى السلسلة النصية التي سنعرضها على الـ Label المخصصة لعرض النتيجة النهائية.

نلاحظ هنا أننا عند المرور على الأسطر المختارة، سنكون بحاجة إلى رقم الطالب لإضافته إلى مجموعة الطلاب المختارة، و يمكننا عمل ذلك من خلال العديد من الطرق و أهمها:

- البحث عن العنصر الذي يحتوي قيمة رقم الطالب بضمن السطر (بنفس الطريقة التي بحثنا فيه عن الـ **chkSelectRow**) و قراءة القيمة.
- الاعتماد على الخاصية **DataKeys** للـ GridView.
- كتابة القيمة المراد قراءتها لاحقاً كـ **Attribute** في العنصر المطلوب (**chkSelectRow** في مثالنا)، ثم قراءة هذه الخاصية برمجياً.

سنعتمد في هذا المثال على الخيار الثالث، و لذلك سنقوم بتعديل بسيط على تعريف العمود الخاص بالـ **CheckBox** ليصبح كالتالي:

```

<asp:CheckBox runat="server" ID="chkSelectRow" StudentId='<%# Eval("Id") %>' />

```

لا يحتوي العنصر **CheckBox** على خاصية اسمها **StudentId**، و لذلك عند كتابتها بهذا الشكل فإن القيمة المعطاة سيتم تخزينها ضمن الـ **Attributes** الخاصة بالـ **CheckBox**، و لذلك يمكن قراءة قيمة هذه الـ **Attribute** برمجياً كما رأينا في المثال السابق من خلال التعليمة:

```

CheckBox c = (CheckBox)row.FindControl("chkSelectRow");
string studentId = c.Attributes["StudentId"];

```

إذا قمنا الآن بتشغيل الصفحة و اختيار بعض الأسطر بشكل عشوائي، ثم الضغط على الزر "عرض" سنحصل على النتيجة التالية:

رقم الطالب	اسم الطالب	معدل	
١	Feras	٦٤	☐
٢	Samer	٩٣	☒
٣	Yamen	٧٢	☒

عرض

لقد قمت باختيار الطلاب التالية:

٢
٣

٨- إضافة CheckBox لاختيار جميع الأسطر دفعةً واحدة:

سنقوم الآن بالتعديل على الكود الخاص بال GridView لإضافة CheckBox في ال Header الخاص بالعمود الأخير، و بالتالي سيصبح تعريف العمود الأخير كما يلي:

```
<asp:TemplateField>
  <HeaderTemplate>
    <asp:CheckBox runat="server" ID="chkCheckAll" Checked="false" />
  </HeaderTemplate>
  <ItemTemplate>
    <asp:CheckBox runat="server" ID="chkSelectRow" StudentId='<%# Eval("Id") %>' />
  </ItemTemplate>
  <ItemStyle Width="60px" HorizontalAlign="Center" />
</asp:TemplateField>
```

رقم الطالب	اسم الطالب	معدل	☐
١	Feras	٦٤	☐
٢	Samer	٩٣	☐
٣	Yamen	٧٢	☐

عرض

سنقوم الآن بكتابة تابع بلغة JavaScript من أجل اختيار جميع الأعمدة (أو إلغاء الاختيار) عند الضغط على ال CheckBox الموجود في الأعلى، لذلك سنكتب التابعين التاليين في قسم ال Head ضمن صفحة ال HTML:

```
<script type="text/javascript" language="javascript">
  function SelectAll(frmId, id, index) {
    var frm = document.getElementById(frmId);

    for (i = 1; i < frm.rows.length; i++) {
      frm.rows[i].cells[index].childNodes[0].childNodes[0].checked =
        document.getElementById(id).checked;
    }
  }

  function UnCheckTop(id) {
    document.getElementById(id).checked = false;
  }
</script>
```

يقوم التابع SelectAll باختيار جميع العناصر الموجودة في الأسطر، و يجب أن نعرف أن ال GridView يتم تحويلها إلى HTML Table و بالتالي يمكن المرور على أسطر ال GridView من خلال حلقة for و باستخدام ال index للوصول إلى عمود معين ضمن السطر.

يقوم التابع UnCheckTop بإلغاء اختيار ال CheckBox الموجودة في ال Header عند إلغاء اختيار أحد العناصر الموجودة في الأسطر.

بقي أمر واحد و هو أن نقوم برمجياً بجعل ال CheckBox الموجودة في كل من الترويسة (Header) و الأسطر بتنفيذ التابع المناسب و تمرير البارامترات المناسبة، و سنقوم بذلك من خلال الحدث RowDataBound لل GridView و الذي يتم تنفيذه بعد تعبئة كل سطر بالبيانات من مصدر البيانات كما يلي:

```
protected void gvTestData_RowDataBound(object sender, GridViewRowEventArgs e)
{
  if ((e.Row.RowType == DataControlRowType.Header))
  {
    //Get the CheckAll checkbox's ClientID to reach it from javascript.
  }
}
```

```

        CheckBox c = (CheckBox)e.Row.FindControl("chkCheckAll");

        //Building the onclick string
        string onclickEvent = string.Format("SelectAll('{0}', '{1}', {2})",
            gvTestData.ClientID, c.ClientID, gvTestData.Columns.Count - 1);

        //registering the event
        c.Attributes.Add("onclick", onclickEvent);
    }
    else if (e.Row.RowType == DataControlRowType.DataRow)
    {
        //Get the CheckAll checkbox's ClientID to reach it from javascript.
        CheckBox c = (CheckBox)gvTestData.HeaderRow.FindControl("chkCheckAll");

        //Get the row checkbox to add the attribute to it.
        CheckBox rowCheckBox = ((CheckBox)e.Row.FindControl("chkSelectRow"));

        //Building the script string.
        string onclickEvent = string.Format("UnCheckTop('{0}']", c.ClientID);

        //Registering the event
        rowCheckBox.Attributes.Add("onclick", onclickEvent);
    }
}

```

نلاحظ ما يلي في الكود السابق:

- العبارة المميزة بالأرضية الصفراء تفترض أن العمود الخاص بال CheckBox هو آخر عمود في ال GridView، إذا لم يكن كذلك يمكننا استبداله بأي رقم نريد، حيث أن هذه القيمة تمثل رقم العمود الخاص بال CheckBox.
- استخدمنا التابع FindControl لأنه لا يمكننا الوصول لل CheckBox مباشرة إذا كان معرفاً ضمن ال GridView.

الآن أصبح الكود الكامل لصفحة ال HTML هو كما يلي:

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Default Page</title>

    <script type="text/javascript" language="javascript">
        function SelectAll(frmId, id, index) {
            var frm = document.getElementById(frmId);

            for (i = 1; i < frm.rows.length; i++) {
                frm.rows[i].cells[index].childNodes[0].checked =
                    document.getElementById(id).checked;
            }
        }

        function UnCheckTop(id) {
            document.getElementById(id).checked = false;
        }
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div dir="rtl">
            <asp:GridView runat="server" ID="gvTestData" AutoGenerateColumns="false"
                OnRowDataBound="gvTestData_RowDataBound">
                <Columns>
                    <asp:BoundField DataField="Id" HeaderText="الطالب رقم">
                        <ItemStyle Width="60px" HorizontalAlign="Center" />
                    </asp:BoundField>
                    <asp:BoundField DataField="Name" HeaderText="الطالب اسم">
                        <ItemStyle Width="160px" HorizontalAlign="Center" />

```

```

        </asp:BoundField>
        <asp:BoundField DataField="Average" HeaderText="معدل">
            <ItemStyle Width="60px" HorizontalAlign="Center" />
        </asp:BoundField>
        <asp:TemplateField>
            <HeaderTemplate>
                <asp:CheckBox runat="server" ID="chkCheckAll"
                    Checked="false" />
            </HeaderTemplate>
            <ItemTemplate>
                <asp:CheckBox runat="server" ID="chkSelectRow"
                    StudentId='<%# Eval("Id") %>' />
            </ItemTemplate>
            <ItemStyle Width="60px" HorizontalAlign="Center" />
        </asp:TemplateField>
    </Columns>
</asp:GridView>
<br />
<asp:Button runat="server" ID="btnGetSelectedStudents" Text="عرض"
    Width="80px" OnClick="btnGetSelectedStudents_Click" />
<br />
<br />
<asp:Label runat="server" ID="lblResult" />
</div>
</form>
</body>
</html>

```

و الكود المكتوب في الـ Code Behind هو:

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!this.IsPostBack)
        {
            gvTestData.DataSource = Student.GetSampleList();
            gvTestData.DataBind();
        }
    }

    protected void btnGetSelectedStudents_Click(object sender, EventArgs e)
    {
        StringBuilder builder = new StringBuilder();

        builder.Append("التالية الطلاب باختيار قمت لقد");
        builder.Append("<br />");

        foreach (GridViewRow row in gvTestData.Rows)
        {
            if (row.RowType == DataControlRowType.DataRow)
            {
                CheckBox c = (CheckBox)row.FindControl("chkSelectRow");
                string studentId = c.Attributes["StudentId"];

                if ((c != null) && (c.Checked == true))
                {
                    builder.Append(studentId);
                    builder.Append("<br />");
                }
            }
        }

        lblResult.Text = builder.ToString();
    }
}

```

```

protected void gvTestData_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if ((e.Row.RowType == DataControlRowType.Header))
    {
        //Get the CheckAll checkbox's ClientID to reach it from javascript.
        CheckBox c = (CheckBox)e.Row.FindControl("chkCheckAll");

        //Building the onclick string
        string onclickEvent = string.Format("SelectAll('{0}', '{1}', {2})",
            gvTestData.ClientID, c.ClientID, gvTestData.Columns.Count - 1);

        //registering the event
        c.Attributes.Add("onclick", onclickEvent);
    }
    else if (e.Row.RowType == DataControlRowType.DataRow)
    {
        //Get the CheckAll checkbox's ClientID to reach it from javascript.
        CheckBox c = (CheckBox)gvTestData.HeaderRow.FindControl("chkCheckAll");

        //Get the row checkbox to add the attribute to it.
        CheckBox rowCheckBox = ((CheckBox)e.Row.FindControl("chkSelectRow"));

        //Building the script string.
        string onclickEvent = string.Format("UnCheckTop('{0}']", c.ClientID);

        //Registering the event
        rowCheckBox.Attributes.Add("onclick", onclickEvent);
    }
}
}

```

يمكننا الآن تجربة الصفحة السابقة بتشغيلها ثم الضغط على الـ CheckBox الموجود في الـ Header، و عندها سيتم اختيار جميع الـ CheckBox الموجودة في الأسطر، بالإضافة إلى إلغاء إختيار الـ CheckBox الموجود في الـ Header عند إلغاء اختيار أحد الأسطر.