Here I will explain how to bind data to asp.net gridview with JQuery or JSON using c#, vb.net or bind data to gridview with JQuery or JSON in asp.net using c#, vb.net.

**Description:**   
     
In previous article I explained [JQuery UI AutoComplete textbox with database](http://www.aspdotnet-suresh.com/2012/03/jquery-ui-autocomplete-textbox-with.html) and [call asp.net pagemethods in JQuery](http://www.aspdotnet-suresh.com/2012/03/how-to-call-aspnet-page-methods-from.html). Now I am using those concepts to explain how to bind data to gridview using JQuery/JSON and display it on aspx page in asp.net.

To implement this concept first we need to design table in database give name as **UserInformation** to save user details in database.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Nulls** |
| UserId | int(set identity property=true) | No |
| UserName | varchar(50) | Yes |
| Location | nvarchar(max) | Yes |

After completion table design enter some of user details in database to work for our sample.

Now create new website and write the following code in your aspx page

|  |
| --- |
| <html xmlns="http://www.w3.org/1999/xhtml">  <head id="Head1" runat="server">  <title>Asp.net Bind Data to Gridview using JQuery or JSON</title>  <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>  <script type="text/javascript">  $(document).ready(function() {  $.ajax({  type: "POST",  contentType: "application/json; charset=utf-8",  url: "Default.aspx/BindDatatable",  data: "{}",  dataType: "json",  success: function(data) {  for (var i = 0; i < data.d.length; i++) {  $("#gvDetails").append("<tr><td>" + data.d[i].UserId + "</td><td>" + data.d[i].UserName + "</td><td>" + data.d[i].Location + "</td></tr>");  }  },  error: function(result) {  alert("Error");  }  });  });  </script>  <style type="text/css">  table,th,td  {  border:1px solid black;  border-collapse:collapse;  }  </style>  </head>  <body>  <form id="form1" runat="server">  <asp:GridView ID="gvDetails" runat="server">  <HeaderStyle BackColor="#DC5807" Font-Bold="true" ForeColor="White" />  </asp:GridView>  </form>  </body>  </html> |

If you observe above code in header section I added script file link by using that file we have a chance to interact with JQuery. If you want to know about script function mentioned in header section check these posts [JQuery UI AutoComplete textbox with database](http://www.aspdotnet-suresh.com/2012/03/jquery-ui-autocomplete-textbox-with.html) and [call asp.net pagemethods in JQuery](http://www.aspdotnet-suresh.com/2012/03/how-to-call-aspnet-page-methods-from.html). Here I explained that function clearly.

Now open code behind file and add the following namespaces

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Web.Services;  using System.Collections.Generic; |

After that write the following code in code behind

**C#.NET Code**

|  |
| --- |
| protected void Page\_Load(object sender, EventArgs e)  {  if (!IsPostBack)  {  BindColumnToGridview();  }  }  // This method is used to bind dummy row to gridview to bind data using JQuery  private void BindColumnToGridview()  {  DataTable dt = new DataTable();  dt.Columns.Add("UserId");  dt.Columns.Add("UserName");  dt.Columns.Add("Location");  dt.Rows.Add();  gvDetails.DataSource = dt;  gvDetails.DataBind();  gvDetails.Rows[0].Visible = false;  }  [WebMethod]  public static UserDetails[] BindDatatable()  {  DataTable dt = new DataTable();  List<UserDetails> details = new List<UserDetails>();  using (SqlConnection con=new SqlConnection("Data Source=SureshDasari;Initial Catalog=MySampleDB;Integrated Security=true"))  {  using (SqlCommand cmd=new SqlCommand("select TOP 10 UserId,UserName,Location from UserInformation",con))  {  con.Open();  SqlDataAdapter da= new SqlDataAdapter(cmd);  da.Fill(dt);  foreach (DataRow dtrow in dt.Rows)  {  UserDetails user=new UserDetails();  user.UserId = dtrow["UserId"].ToString();  user.UserName = dtrow["UserName"].ToString();  user.Location = dtrow["Location"].ToString();  details.Add(user);  }  }  }  return details.ToArray();  }  public class UserDetails  {  public string UserId { get; set; }  public string UserName { get; set; }  public string Location { get; set; }  } |
|  |

