In this video we will discuss implementing **retry logic for deadlock exceptions**.    
  
   
  
This is continuation to [Part 84](http://csharp-video-tutorials.blogspot.com/2015/08/handling-deadlocks-in-adonet.html). Please watch [Part 84](http://csharp-video-tutorials.blogspot.com/2015/08/handling-deadlocks-in-adonet.html), before proceeding.   
  
When a transaction fails due to deadlock, we can write some logic so the system can resubmit the transaction. The deadlocks usually last for a very short duration. So upon resubmitting the transaction it may complete successfully. This is much better from user experience standpoint.  
  
To achieve this we will be using the following technologies  
C#  
ASP.NET  
SQL Server  
jQuery AJAX  
  
**Result.cs**

public class Result

{

    public int AttemptsLeft { get; set; }

    public string Message { get; set; }

    public bool Success { get; set; }

}

**WebForm1.aspx HTML and jQuery code**

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title></title>

    <script src="jquery-1.11.2.js"></script>

    <script type="text/javascript">

        $(document).ready(function () {

            var lblMessage = $('#Label1');

            var attemptsLeft;

            function updateData() {

                $.ajax({

                    url: 'WebForm1.aspx/CallStoredProcedure',

                    method: 'post',

                    contentType: 'application/json',

                    data: '{attemptsLeft:' + attemptsLeft + '}',

                    dataType: 'json',

                    success: function (data) {

                        lblMessage.text(data.d.Message);

                        attemptsLeft = data.d.AttemptsLeft;

                        if (data.d.Success) {

                            $('#btn').prop('disabled', false);

                            lblMessage.css('color','green');

                        }

                        else if(attemptsLeft > 0){

                            lblMessage.css('color', 'red');

                            updateData();

                        }

                        else {

                            lblMessage.css('color', 'red');

                            lblMessage.text('Deadlock Occurred. ZERO attempts left. Please try later');

                        }

                    },

                    error: function (err) {

                        lblMessage.css('color', 'red');

                        lblMessage.text(err.responseText);

                    }

                });

            }

            $('#btn').click(function () {

                $(this).prop('disabled', true);

                lblMessage.text('Updating....');

                attemptsLeft = 5;

                updateData();

            });

        });

    </script>

</head>

<body style="font-family: Arial">

    <form id="form1" runat="server">

        <input id="btn" type="button"

            value="Update Table A and then Table B" />

        <br />

        <asp:Label ID="Label1" runat="server"></asp:Label>

    </form>

</body>

</html>

**WebForm1.aspx.cs code**

using System;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

namespace Demo

{

    public partial class WebForm1 : System.Web.UI.Page

    {

        protected void Page\_Load(object sender, EventArgs e)

        { }

        [System.Web.Services.WebMethod]

        public static Result CallStoredProcedure(int attemptsLeft)

        {

            Result \_result = new Result();

            if (attemptsLeft > 0)

            {

                try

                {

                    string cs =ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;

                    using (SqlConnection con = new SqlConnection(cs))

                    {

                        SqlCommand cmd = new SqlCommand("spTransaction15", con);

                        cmd.CommandType = CommandType.StoredProcedure;

                        con.Open();

                        cmd.ExecuteNonQuery();

                        \_result.Message = "Transaction successful";

                        \_result.AttemptsLeft = 0;

                        \_result.Success = true;

                    }

                }

                catch (SqlException ex)

                {

                    if (ex.Number == 1205)

                    {

                        \_result.AttemptsLeft = attemptsLeft - 1;

                        \_result.Message = "Deadlock occurred. Retrying. Attempts left : "

                            + \_result.AttemptsLeft.ToString();

                    }

                    else

                    {

                        throw;

                    }

                    \_result.Success = false;

                }

            }

            return \_result;

        }

    }

}

Copy and paste the above code in WebForm2.aspx and make the required changes as described in the video.