using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using Microsoft.ApplicationBlocks.Data;

using System.Web.Services;

using System.Data;

using Newtonsoft.Json;

using System.Data.SqlClient;

/// <summary>

/// Summary description for WebService

/// </summary>

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class WebService : System.Web.Services.WebService

{

public static string connString = System.Configuration.ConfigurationManager.ConnectionStrings["connString"].ConnectionString;

[WebMethod]

public void GetTaskList()

{

string sql = "SELECT \* FROM KB\_TASK";

DataTable dt\_Task = ExecuteQueryInDT(connString, sql);

this.Context.Response.ContentType = "application/json; charset=utf-8";

this.Context.Response.Write(JsonConvert.SerializeObject(dt\_Task));

}

[WebMethod]

public void UpdateTaskList(string label, string status)

{

string sql = string.Format(@"UPDATE KB\_TASK SET STATUS = '{0}' WHERE TASKID = '{1}'", status, label);

ExecuteTransactionQuery(connString, sql);

}

public static void ExecuteTransactionQuery(string connString, string sql)

{

string strsql = "BEGIN TRANSACTION; BEGIN TRY ";

strsql += sql + " END TRY BEGIN CATCH SELECT ERROR\_MESSAGE() AS errorMessage; IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION; END CATCH; " +

"IF @@TRANCOUNT > 0 BEGIN COMMIT TRANSACTION; END";

using (SqlConnection connection = new SqlConnection(connString))

{

try

{

connection.Open();

DataSet ds = SqlHelper.ExecuteDataset(connection, CommandType.Text, strsql);

if (ds != null && ds.Tables.Count > 0 && ds.Tables[0] != null && ds.Tables[0].Columns.Count > 0)

{

DataColumn dc = ds.Tables[0].Columns[0];

if (dc.ColumnName == "errorMessage")

if (ds.Tables[0].Rows.Count > 0)

throw new Exception(ds.Tables[0].Rows[0][0].ToString());

}

ds.Dispose();

}

catch (Exception ex)

{

throw new Exception(ex.ToString());

}

}

}

public static DataTable ExecuteQueryInDT(string connString, string strsql)

{

DataTable dt = new DataTable();

using (SqlConnection connection = new SqlConnection(connString))

{

try

{

connection.Open();

DataSet ds = SqlHelper.ExecuteDataset(connection, CommandType.Text, strsql);

if (ds != null && ds.Tables.Count > 0 && ds.Tables[0] != null)

{

dt = ds.Tables[0];

}

ds.Dispose();

}

catch (Exception ex)

{

throw new Exception(ex.ToString());

}

}

return dt;

}

}