How to add a API Header in web api :

using AppBlock;

using FormatFunctions;

using Newtonsoft.Json;

using Newtonsoft.Json.Linq;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace ABCAPI.Controllers

{

public class ABCAPIController : ApiController

{

[HttpPost]

public resultbinddata Marketshare\_price([FromBody] binddata passdata)

{

resultbinddata refe = new resultbinddata();

string storeresult = "";

string Useridhead = "", Usernamehead = "";

DataTable dt = new DataTable();

string[] d = passdata.isin;

dt.Columns.Add("isin");

for (int i = 0; i < d.Length; i++)

{

DataRow dr = dt.NewRow();

dr["isin"] = d[i].ToString();

dt.Rows.Add(dr);

}

try

{

System.Net.Http.Headers.HttpRequestHeaders headers = this.Request.Headers;

if (headers.Contains("clientname"))

{

Usernamehead = headers.GetValues("clientname").First();// header value we can see

}

if (headers.Contains("clientkey"))

{

Useridhead = headers.GetValues("clientkey").First();

}

string sql = " select \* from ABCApiheader where clientname='" + Usernamehead + "' and clientkey='" + Useridhead + "'";

DataSet dshead = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["common165"].ToString(), CommandType.Text, sql);

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

{

#region

if (dt != null && dt.Rows.Count > 0)

{

DataSet ds = new DataSet();

ds.Tables.Add(dt);

string convetXml = ds.GetXml().Replace("\r\n", "").Replace(" ", "");

SqlParameter[] param = { new SqlParameter("@XMLisin", convetXml) };

//SqlParameter[] param = { new SqlParameter("@isin", passdata.isin) };

DataSet ds1 = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["common165"].ToString(), CommandType.StoredProcedure, "pricedetailsBSENSE\_arrayxml", param);

DataView dt4 = ds1.Tables[0].DefaultView;

//storeresult = DataTableToJSONWithJSONNet(dt4.ToTable(true, "isin", "bseprice", "nseprice", "Nsepercentage", "Bsepercentage", "Bsechange", "Nsechange", "B52High", "N52High", "B52LOW", "N52LOW", "BTMCAP", "NTMCAP"));

storeresult = DataTableToJSONWithJSONNet(dt4.ToTable(true, "Isin", "bseprice", "Nseprice", "Nsepercentage", "Bsepercentage", "Bsechange", "Nsechange", "B52High", "N52High", "B52LOW", "N52LOW", "BSEMCAP", "NSEMCAP", "BSEMCAP\_USD", "NSEMCAP\_USD","Company\_Name"));

isindata[] obj = JsonConvert.DeserializeObject<isindata[]>(storeresult);

//JArray jArray = JArray.Parse(storeresult);

if (storeresult != "[]")

{

refe.Status = true;

refe.Message ="Sucessfully Updated";

refe.Data = obj;

return refe;

}

else

{

refe.Status = false;

refe.Message ="No Data Available..!!";

refe.Data = null;

return refe;

}

}

else

{

storeresult = JsonConvert.SerializeObject("ISIN Is Missing..!!");

refe.Status = false;

refe.Message = storeresult;

refe.Data = null;

return refe;

}

#endregion

}

else

{

storeresult = JsonConvert.SerializeObject("Invalid clientname Name or clientkey");

refe.Status = false;

refe.Message = storeresult;

refe.Data = null;

return refe;

}

}

catch (Exception ex)

{

Console.WriteLine("Please try again later: " + ex);

storeresult = JsonConvert.SerializeObject(" No Data Available..!!");

refe.Status = false;

refe.Message = storeresult;

refe.Data = null;

return refe;

}

}

public chartresultbinddata Marketshare\_chart([FromBody] chartRequest chartpassdata)

{

#region 2nd

chartresultbinddata refe = new chartresultbinddata();

string storeresult = string.Empty;

string Useridhead = "", Usernamehead = "";

DataTable dt = new DataTable();

string[] d = chartpassdata.isin.ToArray();

string[] isinArray = chartpassdata.isin.ToArray();

isinArray = isinArray.Where(c => c != null && c != "").ToArray();

int isincount = isinArray.Length;

string isinString = string.Join("','", isinArray);

try

{

System.Net.Http.Headers.HttpRequestHeaders headers = this.Request.Headers;

if (headers.Contains("clientname"))

{

Usernamehead = headers.GetValues("clientname").First();// header value we can see

}

if (headers.Contains("clientkey"))

{

Useridhead = headers.GetValues("clientkey").First();

}

string sql = " select \* from ABCApiheader where clientname='" + Usernamehead + "' and clientkey='" + Useridhead + "'";

DataSet dshead = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["common165"].ToString(), CommandType.Text, sql);

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

{

#region

if (chartpassdata.period != "" && isincount > 0)

{

DataSet ds = new DataSet();

string SQL = "Select co\_code from aliasmas where isin in ('" + isinString + "')";

ds = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["ClineOle"].ToString(), CommandType.Text, SQL);

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

{

string chartco\_code = ds.GetXml().Replace("\r\n", "");

string perioddate = null;

string datef = chartpassdata.period;

if (chartpassdata.period.ToLower().Contains("day") || (chartpassdata.period.ToLower().Contains("week")) || (chartpassdata.period.ToLower().Contains("month")) || (chartpassdata.period.ToLower().Contains("year")))

{

if (chartpassdata.period.ToLower().Contains("day"))

{

#region day

SqlParameter[] param1 = { new SqlParameter("@chartco\_codeXML", chartco\_code), SqlHelper.MakeParam("@Excg", chartpassdata.exchange) };

DataSet ds2 = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["Klsehist"].ToString(), CommandType.StoredProcedure, "cm\_intradaystkprice\_array", param1);

maraketchartdata[] objArr = new maraketchartdata[isinArray.Length];

for (int i = 0; i < isinArray.Length; i++)

{

maraketchartdata obje = new maraketchartdata();

obje.isin = isinArray[i].ToString();

DataTable firstTable = ds2.Tables[0];

DataView dv = firstTable.DefaultView;

dv.RowFilter = "isin='" + obje.isin + "'";

DataTable filtertbl = dv.ToTable();

if (filtertbl != null && filtertbl.Rows.Count > 0)

{

storeresult = DataTableToJSONWithJSONNet2(filtertbl);

dataarray[] isindataArr = JsonConvert.DeserializeObject<dataarray[]>(storeresult);

obje.isinData = isindataArr;

}

else

obje.isinData = null;

objArr[i] = obje;

}

refe.Status = true;

refe.Message = "Sucessfully Updated";

refe.Data = objArr;

return refe;

#endregion

}

else

{

#region Others

if (chartpassdata.period.ToLower().Contains("week"))

{

datef = datef.ToLower().Replace("week", "");

perioddate = C\_Format.M\_FormatDate(DateTime.Now.AddDays(-(7 \* (Convert.ToInt32(datef)))).ToShortDateString(), "yyyy-MM-dd");

}

if (chartpassdata.period.ToLower().Contains("month"))

{

datef = datef.ToLower().Replace("month", "");

perioddate = C\_Format.M\_FormatDate(DateTime.Now.AddMonths(-(Convert.ToInt32(datef))).ToShortDateString(), "yyyy-MM-dd");

}

if (chartpassdata.period.ToLower().Contains("year"))

{

datef = datef.ToLower().Replace("year", "");

perioddate = C\_Format.M\_FormatDate(DateTime.Now.AddYears(-(Convert.ToInt32(datef))).ToShortDateString(), "yyyy-MM-dd");

}

SqlParameter[] param = { new SqlParameter("@chartco\_codeXML", chartco\_code), SqlHelper.MakeParam("@exchg", chartpassdata.exchange), new SqlParameter("@EndDate", perioddate) };

DataSet ds1 = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["common165"].ToString(), CommandType.StoredProcedure, "Marketshare\_charts\_array", param);

maraketchartdata[] objArr = new maraketchartdata[isinArray.Length];

for (int i = 0; i < isinArray.Length; i++)

{

maraketchartdata obje = new maraketchartdata();

obje.isin = isinArray[i].ToString();

DataTable firstTable = ds1.Tables[0];

DataView dv = firstTable.DefaultView;

dv.RowFilter = "isin='" + obje.isin + "'";

DataTable filtertbl = dv.ToTable();

if (filtertbl != null && filtertbl.Rows.Count > 0)

{

storeresult = DataTableToJSONWithJSONNet2(filtertbl);

dataarray[] isindataArr = JsonConvert.DeserializeObject<dataarray[]>(storeresult);

obje.isinData = isindataArr;

}

else

obje.isinData = null;

objArr[i] = obje;

}

if (storeresult != "[]")

{

refe.Status = true;

refe.Message = "Sucessfully Updated";

refe.Data = objArr;

return refe;

}

else

{

refe.Status = false;

refe.Message ="No Data Available..!!";

refe.Data = null;

return refe;

}

#endregion

}

}

else

{

refe.Status = false;

refe.Message = JsonConvert.SerializeObject(" Invalid Period..!!");

refe.Data = null;

return refe;

}

}

else

{

storeresult = JsonConvert.SerializeObject("Invalid ISIN ..!!");

refe.Status = false;

refe.Message = storeresult;

refe.Data = null;

return refe;

}

}

else

{

storeresult = JsonConvert.SerializeObject("ISIN or Period Is Missing..!!");

refe.Status = false;

refe.Message = storeresult;

refe.Data = null;

return refe;

}

#endregion

}

else

{

storeresult = JsonConvert.SerializeObject("Invalid clientname Name or clientkey");

refe.Status = false;

refe.Message = storeresult;

refe.Data = null;

return refe;

}

}

catch (Exception ex)

{

Console.WriteLine("Please try again later: " + ex);

storeresult = JsonConvert.SerializeObject(" No Data Available..!!");

refe.Status = false;

refe.Message = storeresult;

refe.Data = null;

return refe;

}

#endregion

}

#region Companyfinancial\_performance

//public resultbinddata Companyfinancial\_performance([FromBody] cmpyperformance passdata)

//{

// #region finall

// resultbinddata refe = new resultbinddata();

// string storeresult = "";

// string cocode = "";

// string Useridhead = "", Usernamehead = "";

// try

// {

// System.Net.Http.Headers.HttpRequestHeaders headers = this.Request.Headers;

// if (headers.Contains("clientname"))

// {

// Usernamehead = headers.GetValues("clientname").First();// header value we can see

// }

// if (headers.Contains("clientkey"))

// {

// Useridhead = headers.GetValues("clientkey").First();

// }

// string sql = " select \* from ABCApiheader where clientname='" + Usernamehead + "' and clientkey='" + Useridhead + "'";

// DataSet dshead = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["common165"].ToString(), CommandType.Text, sql);

// if (dshead != null && dshead.Tables[0].Rows.Count > 0)

// {

// #region

// if (passdata.isin != "" && passdata.period != "" && passdata.performance != "")

// {

// DataSet ds = new DataSet();

// string SQL = "Select co\_code from aliasmas where isin='" + passdata.isin + "'";

// ds = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["ClineOle"].ToString(), CommandType.Text, SQL);

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

// {

// cocode = ds.Tables[0].Rows[0]["Co\_Code"].ToString();

// if ((passdata.period.ToLower() == "quarterly" || passdata.period.ToLower() == "yearly") && (passdata.performance.ToLower() == "standalone" || passdata.performance.ToLower() == "consolidated"))

// {

// SqlParameter[] param = { new SqlParameter("@co\_code", cocode),

// new SqlParameter("@period", passdata.period),

// new SqlParameter("@performance", passdata.performance),

// };

// DataSet ds1 = SqlHelper.ExecuteDataset(ConfigurationManager.ConnectionStrings["common165"].ToString(), CommandType.StoredProcedure, "Companyfinancial\_performance", param);

// //DataView dt4 = ds1.Tables[0].DefaultView;

// //storeresult = DataTableToJSONWithJSONNet(dt4.ToTable("", "", ""));

// storeresult = DataTableToJSONWithJSONNet(ds1.Tables[0]);

// if (storeresult != "[]")

// {

// refe.Status = true;

// refe.Message = JsonConvert.SerializeObject("Sucessfully Updated");

// refe.Data = storeresult;

// return refe;

// }

// else

// {

// refe.Status = false;

// refe.Message = JsonConvert.SerializeObject("No Data Available..!!");

// refe.Data = null;

// return refe;

// }

// }

// else

// {

// storeresult = JsonConvert.SerializeObject("Invalid Period or Performance..!!");

// refe.Status = false;

// refe.Message = storeresult;

// refe.Data = null;

// return refe;

// }

// }

// else

// {

// storeresult = JsonConvert.SerializeObject("Invalid ISIN..!!");

// refe.Status = false;

// refe.Message = storeresult;

// refe.Data = null;

// return refe;

// }

// }

// else

// {

// storeresult = JsonConvert.SerializeObject("ISIN or Period or Performance Is Missing..!!");

// refe.Status = false;

// refe.Message = storeresult;

// refe.Data = null;

// return refe;

// }

// #endregion

// }

// else

// {

// storeresult = JsonConvert.SerializeObject("Invalid clientname Name or clientkey");

// refe.Status = false;

// refe.Message = storeresult;

// refe.Data = null;

// return refe;

// }

// }

// catch (Exception ex)

// {

// Console.WriteLine("Please try again later: " + ex);

// storeresult = JsonConvert.SerializeObject(" No Data Available..!!");

// refe.Status = false;

// refe.Message = storeresult;

// refe.Data = null;

// return refe;

// }

// #endregion

//}

#endregion

public string DataTableToJSONWithJSONNet(DataTable table)

{

string JSONString = string.Empty;

//JSONString = JsonConvert.SerializeObject(table,Newtonsoft.Json.Formatting.Indented);

JSONString = JsonConvert.SerializeObject(table);

return JSONString;

}

public string DataTableToJSONWithJSONNet1(DataTable table)

{

string JSONString = string.Empty;

table.Columns.Remove("Volume");

JSONString = JsonConvert.SerializeObject(table);

//JSONString = JsonConvert.SerializeObject(table,Newtonsoft.Json.Formatting.Indented);

return JSONString;

}

public string DataTableToJSONWithJSONNet2(DataTable table)

{

string JSONString = string.Empty;

table.Columns.Remove("isin");

JSONString = JsonConvert.SerializeObject(table);

return JSONString;

}

public class resultbinddata

{

public Boolean Status { get; set; }

public string Message { get; set; }

public isindata[] Data { get; set; }

}

public class chartresultbinddata

{

public Boolean Status { get; set; }

public string Message { get; set; }

//public chartdata[] Data { get; set; }

public maraketchartdata[] Data { get; set; }

}

public class maraketchartdata

{

public string isin { get; set; }

public dataarray[] isinData { get; set; }

}

public class dataarray

{

public string ClosePrice { get; set; }

public string Date { get; set; }

}

public class isindata

{

public string Isin { get; set; }

public string Bseprice { get; set; }

public string NsePrice { get; set; }

public string Nsepercentage { get; set; }

public string Bsepercentage { get; set; }

public string Bsechange { get; set; }

public string Nsechange { get; set; }

public string B52HIGH { get; set; }

public string N52HIGH { get; set; }

public string B52LOW { get; set; }

public string N52LOW { get; set; }

public string BSEMCAP { get; set; }

public string NSEMCAP { get; set; }

public string BSEMCAP\_USD { get; set; }

public string NSEMCAP\_USD { get; set; }

public string Company\_Name { get; set; }

}

public class chartdata

{

public string ISIN { get; set; }

public string ClosePrice { get; set; }

public string Date { get; set; }

}

//public class inputparameter

//{

// public List<binddata> parameter { get; set; }

//}

public class binddata

{

public string[] isin { get; set; }

}

public class cmpyperformance

{

public string isin { get; set; }

public string period { get; set; }

public string performance { get; set; }

// public List<particular> parameter { get; set; }

}

public class particular

{

public string Revenue { get; set; }

public string NetProfit { get; set; }

public string NetWorth { get; set; }

}

public class chartbinddata

{

public string isin { get; set; }

public string period { get; set; }

public string exchange { get; set; }

}

public class chartParameter

{

public List<string> isin { get; set; }

public string period { get; set; }

public string exchange { get; set; }

}

public class chartRequest

{

// public List<chartParameter> Parameter { get; set; }

public List<string> isin { get; set; }

public string period { get; set; }

public string exchange { get; set; }

}

}

}

How to add Authorization in api :

public static string ValideUPID(string Upi)

{

string pRequestJson = "";

upidvalid req = new upidvalid();

req.vpa = Upi;

string reqjson = (new JavaScriptSerializer()).Serialize(req);

pRequestJson = reqjson;

string lServiceUrl = "https://api.razorpay.com/v1/payments/validate/vpa";

BusinessData.NSE\_ExceptionLogging.NSE\_SendErrorToText("ValideUPID Url:" + lServiceUrl);

#region

string loutpout\_Response = "";

HttpWebRequest lhttpRequest = null;

try

{

string status = "", cname = "", vpa = "", reason = "", code = "", description = "";

string userName = "rzp\_live\_VAA0SRBgMbAHjn", passWord = "Ity1RvccLFEeF1qFW2BQ4Oah";

ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls | SecurityProtocolType.Ssl3 | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls12;

ServicePointManager.ServerCertificateValidationCallback = delegate (object sender, System.Security.Cryptography.X509Certificates.X509Certificate pCertificate, System.Security.Cryptography.X509Certificates.X509Chain pChain, System.Net.Security.SslPolicyErrors pSSLPolicyErrors) { return true; };

lhttpRequest = (HttpWebRequest)WebRequest.Create(new Uri(lServiceUrl));

lhttpRequest.Headers.Add("Authorization", "Basic " + Convert.ToBase64String(System.Text.Encoding.Default.GetBytes(userName + ":" + passWord)));

lhttpRequest.Accept = "application/json";

lhttpRequest.ContentType = "application/json";

lhttpRequest.KeepAlive = true;

lhttpRequest.Method = "POST";

byte[] bytes1 = Encoding.UTF8.GetBytes(pRequestJson);

using (Stream stream = lhttpRequest.GetRequestStream())

{

stream.Write(bytes1, 0, bytes1.Length);

stream.Close();

}

BusinessData.NSE\_ExceptionLogging.NSE\_SendErrorToText("ValideUPID Request:" + pRequestJson);//log

using (HttpWebResponse httpResponse = (HttpWebResponse)lhttpRequest.GetResponse())

{

using (Stream strm = httpResponse.GetResponseStream())

{

loutpout\_Response = (new StreamReader(strm)).ReadToEnd();

var jsonobj = new JavaScriptSerializer().Deserialize<Dictionary<string, string>>(loutpout\_Response);

status = jsonobj["success"].ToString();

if (status.ToLower() == "true")

{

cname = jsonobj["customer\_name"].ToString();

vpa = jsonobj["vpa"].ToString();

BusinessData.NSE\_ExceptionLogging.NSE\_SendErrorToText(" ValideUPID customer\_name :" + cname);

return cname;

}

else {

BusinessData.NSE\_ExceptionLogging.NSE\_SendErrorToText(" ValideUPID Error status :" + description);

return "failed";

}

}

}

}

catch (Exception ex)

{

string str = "Request:- " + pRequestJson + " \n Response:-" + loutpout\_Response.ToString();

BusinessData.NSE\_ExceptionLogging.NSE\_SendErrorToText("ValideUPID Error:" + ex.Message.ToString());

return "error - " + ex.Message.ToString();

}

#endregion

}

AJAX call :

Basic Syntax:

$.ajax({

type: "POST",

contentType: "application/json; charset=utf-8",

url: "/Research/Research.aspx/filldata",

data: "{type:'" + SubType + "',calls:'false'}",

dataType: "json",

success: function (output) {

var s = output.d;

if (s == '') {

$('#NoData').show();

$('#NoData').html("No Data Available")

$('#researchpdfdiv').hide();

} else {

$('#NoData').hide();

$('#researchpdfdiv').html(s);

$('#researchpdfdiv').show();

}

},

error: function (errormsg) {

}

});

Connection string :

<connectionStrings>

<add name="DefaultConnection" connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-ABCAPI-20240322125229.mdf;Initial Catalog=aspnet-ABCAPI-20240322125229;Integrated Security=True" providerName="System.Data.SqlClient" />

<add name="common165" connectionString="Data Source=SoftSQL;Initial Catalog=CommonDb;Connect TimeOut=60; Max Pool Size=10000 ;user id=sa;password=capmark@09" providerName="System.Data.SqlClient" />

<add name="ClineOle" connectionString="Data Source=SoftSQL;Initial Catalog=ClineOle;Connect TimeOut=60; Max Pool Size=10000 ;user id=sa;password=capmark@09" providerName="System.Data.SqlClient" />

<add name="Klsehist" connectionString="Data Source=SoftSQL;Initial Catalog=Klsehist;Connect TimeOut=60; Max Pool Size=10000 ;user id=sa;password=capmark@09" providerName="System.Data.SqlClient" />

</connectionStrings>

Ajax call and split and retrieve data

[WebMethod]

public static string Retrievecategory(string catsymbol, string exc)

{

string sqlcon = ConfigurationManager.ConnectionStrings["IPOOnline"].ToString();

string sqlcat = "";

if (exc == "NSE")

{

//sqlcat = "select DISTINCT A.CStatus,A.Category from IPOCategoryMaster a inner join nse\_ipomaster b on a.CStatus=b.IPO\_cat WHERE B.IPO\_SymBse='" + catsymbol + "'";

sqlcat = "select DISTINCT A.CStatus,A.Category from IPOCategoryMaster a inner join nse\_ipomaster b on a.CStatus=b.IPO\_cat WHERE B.IPO\_SymBse=@catsymbol";

}

else

{

//sqlcat = "select DISTINCT A.CStatus,A.Category from IPOCategoryMaster a inner join ipomaster b on a.CStatus=b.IPO\_cat WHERE B.IPO\_SymBse='" + catsymbol + "'";

sqlcat = "select DISTINCT A.CStatus,A.Category from IPOCategoryMaster a inner join ipomaster b on a.CStatus=b.IPO\_cat WHERE B.IPO\_SymBse=@catsymbol";

}

SqlParameter[] param = { new SqlParameter("@catsymbol", catsymbol) };

DataSet catds = SqlHelper.ExecuteDataset(sqlcon, CommandType.Text, sqlcat, param);

string result = null;

if (catds != null)

{

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

{

for (int i = 0; i < catds.Tables[0].Rows.Count; i++)

{

result += catds.Tables[0].Rows[i]["Category"].ToString().Trim() + "~" + catds.Tables[0].Rows[i]["CStatus"].ToString().Trim() + "|";

}

result = result.TrimEnd('|');

}

}

return result;

}

2) second one

[WebMethod]

public static string fillipocategory\_discount(string Category, string Symbol, string exc)

{

string sqlcon = ConfigurationManager.ConnectionStrings["IPOOnline"].ToString();

string strsql = "";

if (exc == "NSE")

{

// strsql = "select ipo\_discount,ipo\_discount\_type from nse\_ipomaster where ipo\_symbse ='" + Symbol + "' and cast(ipo\_cat as varchar(50)) = '" + Category + "'";

strsql = "select ipo\_discount,ipo\_discount\_type from nse\_ipomaster where ipo\_symbse =@Symbol and cast(ipo\_cat as varchar(50)) = @Category";

}

else

{

//strsql = "select ipo\_discount,ipo\_discount\_type from ipomaster where ipo\_symbse ='" + Symbol + "' and cast(ipo\_cat as varchar(50)) = '" + Category + "'";

strsql = "select ipo\_discount,ipo\_discount\_type from ipomaster where ipo\_symbse =@Symbol and cast(ipo\_cat as varchar(50)) = @Category";

}

SqlParameter[] param = { new SqlParameter("@Symbol", Symbol),

new SqlParameter("@Category", Category) };

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

string result = null;

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

{

if (ds.Tables[0].Rows[0]["ipo\_discount\_type"].ToString() == "3" || ds.Tables[0].Rows[0]["ipo\_discount\_type"].ToString() == "P")

{

result = "(in %) :";

}

else

{

result = "(in ₹ ) :";

}

result += '|' + (ds.Tables[0].Rows[0]["ipo\_discount"].ToString().Trim());

}

return result;

}

Front end content :

function categorysymbol() {

//debugger;

var symbol = document.getElementById("symbolhdn").value;

$.ajax({

type: "POST",

contentType: "application/json; charset=utf-8",

data: "{'catsymbol':'" + symbol + "','exc':'" + document.getElementById("exchanges").value + "'}",

url: "/ipo.aspx/Retrievecategory",

dataType: "json",

success: function (data) {

// debugger;

var userData = data.d;

// alert(userData)

var value1 = userData.split('|')

var select = document.getElementById("category\_drp");

//select.add(option, "hni");

select.length = 0;

for (var i = 0; i < value1.length; i++) {

var value2 = value1[i].split('~');

var option = document.createElement('option');

//if (i == 0)

// $("#categoryhdn").val(value2[1]);

// alert(value2[1])

option.text = value2[0];

option.value = value2[1];

select.add(option, i);

if (option.value == "IND") {

option.selected = true;

}

//select.add(option, "HNI");

//option.value = "IND";

//option.text = "HNI";

}

//var option = document.createElement('option');

//option.text = HNI;

//option.value = IND;

//select.add(option, value1.length);

select.options.add(new Option('HNI', 'IND'))

categorydiscount();

},

error: function (result) {

alertify.alert(result.responseText);

}

});

}

function categorydiscount() {

//debugger;

//var res = document.getElementById("category\_drp").value;

//alert(res);

// alert("{'Category':'" + document.getElementById("category\_drp").value + "','Symbol':'" + document.getElementById("symbolhdn").value + "'}")

$.ajax({

type: "POST",

contentType: "application/json; charset=utf-8",

data: "{'Category':'" + document.getElementById("category\_drp").value + "','Symbol':'" + document.getElementById("symbolhdn").value + "','exc':'" + document.getElementById("exchanges").value + "'}",

url: "/ipo.aspx/fillipocategory\_discount",

dataType: "json",

success: function (data) {

// debugger;

var userData = data.d;

var ab = document.getElementById("category\_drp");

// alert(ab.options[ab.selectedIndex].text)

var category = ab.options[ab.selectedIndex].text;

var value1 = userData.split('|');

//var res = document.getElementById("lbldiscount1").style.font="14px";

//var res2 = document.getElementById("lbldiscount2");

//var res3 = document.getElementById("lbldiscount3");//discount label id

//res.innerHTML ="Discfgfdsgount : ₹"+ (value1[1])+'&nbsp;'+"per share";

//res2.innerHTML = "Discount : ₹" + (value1[1]) + '&nbsp;' + "per share";

//res3.innerHTML = "Discount : ₹" + (value1[1]) + '&nbsp;' + "per share";

//alert(value1[1])

//document.getElementById("lbldiscount1").innerHTML = "Discount : ₹" + (value1[1]) + '&nbsp;' + "per share";

//document.getElementById("lbldiscount2").innerHTML = "Discount : ₹" + (value1[1]) + '&nbsp;' + "per share";

//document.getElementById("lbldiscount3").innerHTML = "Discount : ₹" + (value1[1]) + '&nbsp;' + "per share";

document.getElementById("lbldiscount1").innerHTML = "Discount " + value1[0] + '&nbsp;' + (value1[1]) + '&nbsp;' + "per share";

document.getElementById("lbldiscount2").innerHTML = "Discount " + value1[0] + '&nbsp;' + (value1[1]) + '&nbsp;' + "per share";

document.getElementById("lbldiscount3").innerHTML = "Discount " + value1[0] + '&nbsp;' + (value1[1]) + '&nbsp;' + "per share";

$("#discounthdn").val(value1[1]);

if (category == "HNI") {

//alert(category);

var minprice = document.getElementById("minprice").value;

var maxprice = document.getElementById("maxprice").value;

var lot = document.getElementById("minqty").value;

var txtTxt = 200000 / (maxprice \* lot);

var lotvalue1 = document.getElementById("<%=lot1.ClientID%>").value = Math.ceil(txtTxt);

var pricevalue1 = document.getElementById("<%=price1.ClientID%>").value = maxprice;

closebid2();

closebid3();

}

else {

var MaxP = parseFloat($("#maxprice").val())

var minq = parseFloat($("#minqty").val())

document.getElementById("<%=lot1.ClientID%>").value = "1";

document.getElementById('<%=cutoff1.ClientID%>').checked = true;

<%-- if (cutoff1.checked != false) {

var minprice = document.getElementById("minprice").value;

//var pricevalue1 = document.getElementById("<%=price1.ClientID%>").value = minprice;

var pricevalue1 = document.getElementById("<%=price2.ClientID%>").value = minprice;

var pricevalue1 = document.getElementById("<%=price3.ClientID%>").value = minprice;

}

alert(cutoff1.checked)

if (cutoff1.checked != true) {

alert(cutoff1.checked)

var minprice = document.getElementById("minprice").value;

var pricevalue1 = document.getElementById("<%=price1.ClientID%>").value = minprice;

}--%>

chk('cutoff1');

closebid2();

closebid3();

$("#addbtn").hide();

}

<%-- else if (category != "HNI") {

//alert(category);

document.getElementById("lot1").value = "1";

document.getElementById("lot2").value = "1";

document.getElementById("lot3").value = "1";

document.getElementById("<%=price1.ClientID%>").value = "";

document.getElementById("<%=price2.ClientID%>").value = "";

document.getElementById("<%=price3.ClientID%>").value = "";

document.getElementById('<%=cutoff1.ClientID%>').checked = false;

document.getElementById("<%=total.ClientID%>").innerHTML = "₹ 0.00 ";

$('#category\_drp').change(function () {

$('#bidhead1').show();

$('#bidhead2').hide();

$('#bidhead3').hide();

});

$("#addbtn").show();

}--%>

CalculateAll();

},

error: function (result) {

alertify.alert(result.responseText);

}

});

}