



链滴

# C#EXCEL 导入导出

作者: [SDX97](#)

原文链接: <https://ld246.com/article/1606148126451>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 导入

```
<a href="javascript:void(0)" data-toggle="topjui-menubutton" data-options="event:'import'"
onclick="imp()">导入外部数据</a>
//js
function imp() {
    var file = $("#file").val();
    var year = document.getElementById("year").value;
    var month = document.getElementById("month").value;
    if (file == null || file.length == 0) {
        alert("请先选择上传文件! ");
        return false;
    }
    $('#file-form').ajaxSubmit({
        url: "../controller/CrmCon.ashx?action=importAreaSimulatedProfit&year=" + year +
&month=" + month + "",
        type: "post",
        datatype:'text',
        success: function (data) {
            //返回根据需求可省略
        }
    });
}
```

## 后台代码

```
HttpPostedFile filePost = context.Request.Files[0];
DataTable myDataTable=SaveExcelFile(filePost);
public static DataTable SaveExcelFile(HttpPostedFile file)
{
    try
    {
        var fileName = file.FileName.Insert(file.FileName.LastIndexOf('.'), "-" + DateTime.No
.ToString("yyyyMMddHHmmssfff"));
        string fileExtName = file.FileName.Substring(file.FileName.LastIndexOf(".") + 1);
        var filePath = Path.Combine(HttpContext.Current.Server.MapPath("../upload"), fileN
me);

        string directoryName = Path.GetDirectoryName(filePath);
        //创建目录
        if (!Directory.Exists(directoryName))
        {
            Directory.CreateDirectory(directoryName);
        }
        if (File.Exists(HttpContext.Current.Server.MapPath("../upload/" + fileName + "")))
        {
            File.Delete(HttpContext.Current.Server.MapPath("../upload/" + fileName + ""));
        }
        file.SaveAs(filePath);
        //HDR=Yes, 这代表第一行是标题, 不做为数据使用, 如果用HDR=NO, 则表示第一行
是标题, 做为数据来使用。系统默认的是YES
        string connstr2003 = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + HttpCo
text.Current.Server.MapPath("../upload/" + fileName + "") + ";Extended Properties='Excel 8.0;H
R=Yes;IMEX=1;";
```

```

        string connstr2007 = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + HttpContext.Current.Server.MapPath("../upload/" + fileName + ".xls") + ";Extended Properties=Excel 12.0;DR=YES\",";
        OleDbConnection conn;
        OleDbDataAdapter myCommand;
        string sql = "select * from [Sheet1$]";
        if (fileExtName == ".xls")
        {
            conn = new OleDbConnection(connstr2003);
            myCommand = new OleDbDataAdapter(sql, connstr2003);
        }
        else
        {
            conn = new OleDbConnection(connstr2007);
            myCommand = new OleDbDataAdapter(sql, connstr2007);
        }
        conn.Open();
        DataTable myDatatable = new DataTable();
        myCommand.Fill(myDatatable);
        conn.Close();
        if (File.Exists(HttpContext.Current.Server.MapPath("../upload/" + fileName + ".xls")))
        {
            File.Delete(HttpContext.Current.Server.MapPath("../upload/" + fileName + ".xls"));
        }
        return myDatatable;
    }
    catch
    {
        return null;
    }
}

```

## 导出

```

<a href="javascript:void(0)" data-toggle="topjui-menubutton" data-options="event:'export'"
onclick="exp()">导出EXCEL</a>
//js
function exp() {
    var year = document.getElementById("year").value;
    var month = document.getElementById("month").value;
    window.open("../controller/CrmCon.ashx?action=ExportQueryAreaSimulatedProfit&year=" + year + "&month=" + month + "");
}

```

## 后台代码

```

public void 导出方法(){
    string fileName = "xxxx表_" + DateTime.Now.ToString("yyyyMMddHHmmss");
    string[] colListHead = new string[]
    {
        "EXCEL列名", "XXXX", "XXX"
    };
    string[] colList = new string[]
    {

```

```

"与之对应的查询出的数据源列名","XXXX", "XXX"
};
System.Data.DataTable dt = Connect.CrmAop.QueryAreaSimulatedProfit(year, mont
);

string rpt_name = "sheet1";
HSSFWorkbook hSSFWorkbook = new HSSFWorkbook();
System.Collections.Generic.List<HSSFCellStyle> list = new System.Collections.Gener
c.List<HSSFCellStyle>();
int num = 0;
HSSFSheet hssfBase = GetHssfBase(hSSFWorkbook, list, rpt_name+num.ToString() );
HSSFRow hSSFRow = (HSSFRow)hssfBase.CreateRow(0);
int num2 = 0;
int num3 = 0;
hSSFRow = (HSSFRow)hssfBase.CreateRow(num2);
for (int i = 0; i < colListHead.Length; i++)
{
    string text = colListHead[i];
    hSSFRow.CreateCell(num3).SetCellValue(text);
    hSSFRow.Cells[num3].CellStyle = list[0];
    num3++;
}
foreach (System.Data.DataRow dataRow in dt.Rows)
{
    num2++;

    hSSFRow = (HSSFRow)hssfBase.CreateRow(num2);
    int num4 = 0;
    for (int i = 0; i < colList.Length; i++)
    {
        string text = colList[i];
        hSSFRow.CreateCell(num4);
        hSSFRow.Cells[num4].SetCellValue(dataRow[text].ToString());
        hSSFRow.Cells[num4].CellStyle = list[1];
        num4++;
    }
}
HttpResponse httpResponse = HttpContext.Current.Response;
httpResponse.Clear();
httpResponse.Buffer = true;
httpResponse.Charset = Encoding.UTF8.BodyName;
httpResponse.AppendHeader("Content-Disposition", "attachment;filename=" + file
ame + ".xls");
httpResponse.ContentEncoding = Encoding.UTF8;
httpResponse.ContentType = "application/vnd.ms-excel;";
hSSFWorkbook.Write(httpResponse.OutputStream);
httpResponse.End();
}
//导出Sheet
private static HSSFSheet GetHssfBase(HSSFWorkbook hssfworkbook, List<HSSFCellStyle
styleList, string rpt_name)
{
    HSSFCellStyle hSSFCellStyle = (HSSFCellStyle)hssfworkbook.CreateCellStyle();
    HSSFFont hSSFFont = (HSSFFont)hssfworkbook.CreateFont();

```

```

hSSFCellStyle.SetFont(hSSFFont);
hSSFCellStyle.Alignment = NPOI.SS.UserModel.HorizontalAlignment.Center;
hSSFCellStyle.VerticalAlignment = VerticalAlignment.Center;
hSSFFont.FontHeightInPoints = 10;
hSSFFont.FontName = "Arial";
hSSFFont.Boldweight = 700;
hSSFCellStyle.FillForegroundColor = 22;
hSSFCellStyle.FillPattern = FillPattern.SolidForeground;
hSSFCellStyle.BorderLeft = NPOI.SS.UserModel.BorderStyle.Thin;
hSSFCellStyle.BorderRight = NPOI.SS.UserModel.BorderStyle.Thin;
hSSFCellStyle.BorderTop = NPOI.SS.UserModel.BorderStyle.Thin;
hSSFCellStyle.BorderBottom = NPOI.SS.UserModel.BorderStyle.Thin;
styleList.Add(hSSFCellStyle);
hSSFCellStyle = (HSSFCellStyle)hssfworkbook.CreateCellStyle();
hSSFFont = (HSSFFont)hssfworkbook.CreateFont();
hSSFCellStyle.SetFont(hSSFFont);
hSSFCellStyle.Alignment = NPOI.SS.UserModel.HorizontalAlignment.Left;
hSSFCellStyle.VerticalAlignment = VerticalAlignment.Center;
hSSFFont.FontHeightInPoints = 9;
hSSFFont.FontName = "Arial";
hSSFFont.Boldweight = 400;
hSSFCellStyle.BorderLeft = NPOI.SS.UserModel.BorderStyle.Thin;
hSSFCellStyle.BorderRight = NPOI.SS.UserModel.BorderStyle.Thin;
hSSFCellStyle.BorderTop = NPOI.SS.UserModel.BorderStyle.Thin;
hSSFCellStyle.BorderBottom = NPOI.SS.UserModel.BorderStyle.Thin;
styleList.Add(hSSFCellStyle);
HSSFSheet hSSFSheet = (HSSFSheet)hssfworkbook.CreateSheet(rpt_name);
hSSFSheet.DefaultColumnWidth = 14;
return hSSFSheet;
}

```