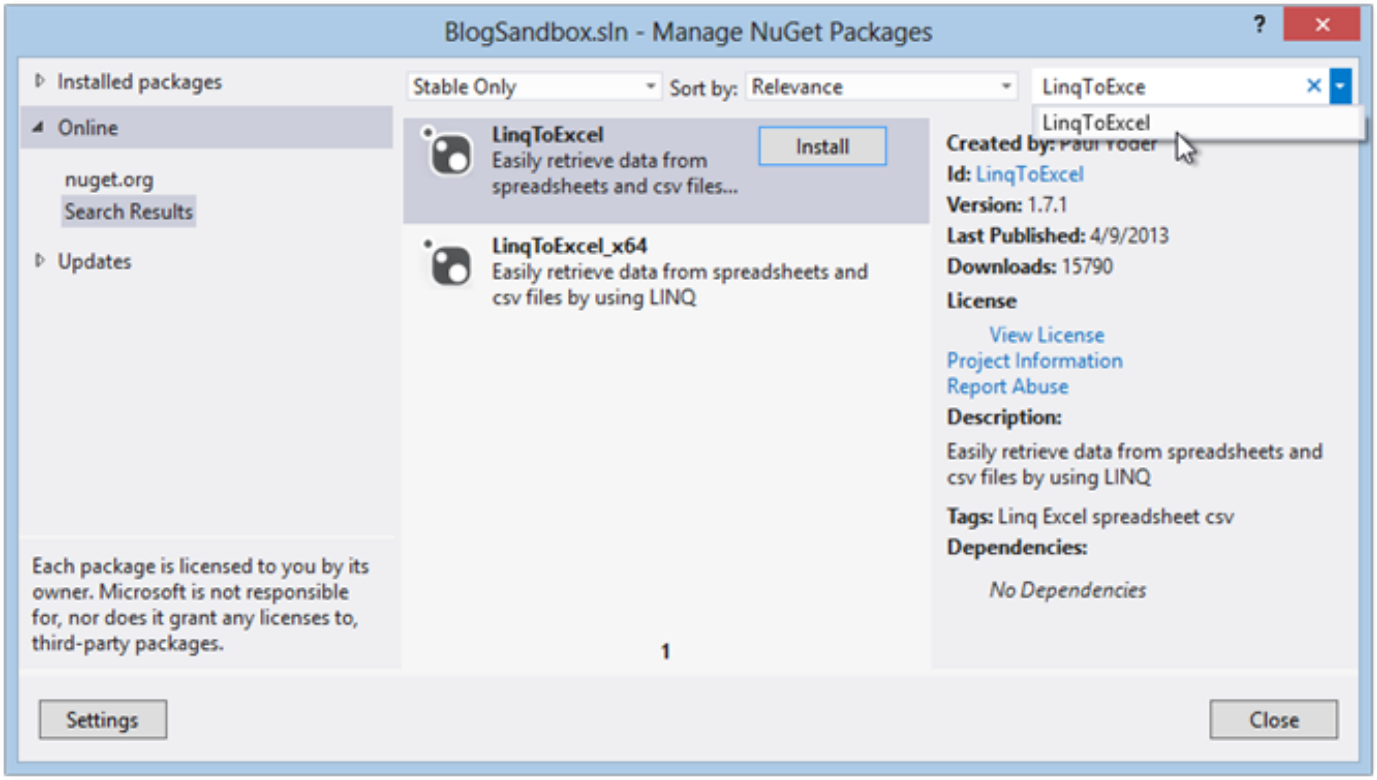
عنوان: **خواندن اطلاعات از فایل اکسل با استفاده از LinqToExcel**
نویسنده: مسعود حق شناس
تاریخ: ۱۳۹۲/۰۸/۰۶ ۲۲:۰
آدرس: www.dotnettips.info
برچسب‌ها: LINQ, Excel

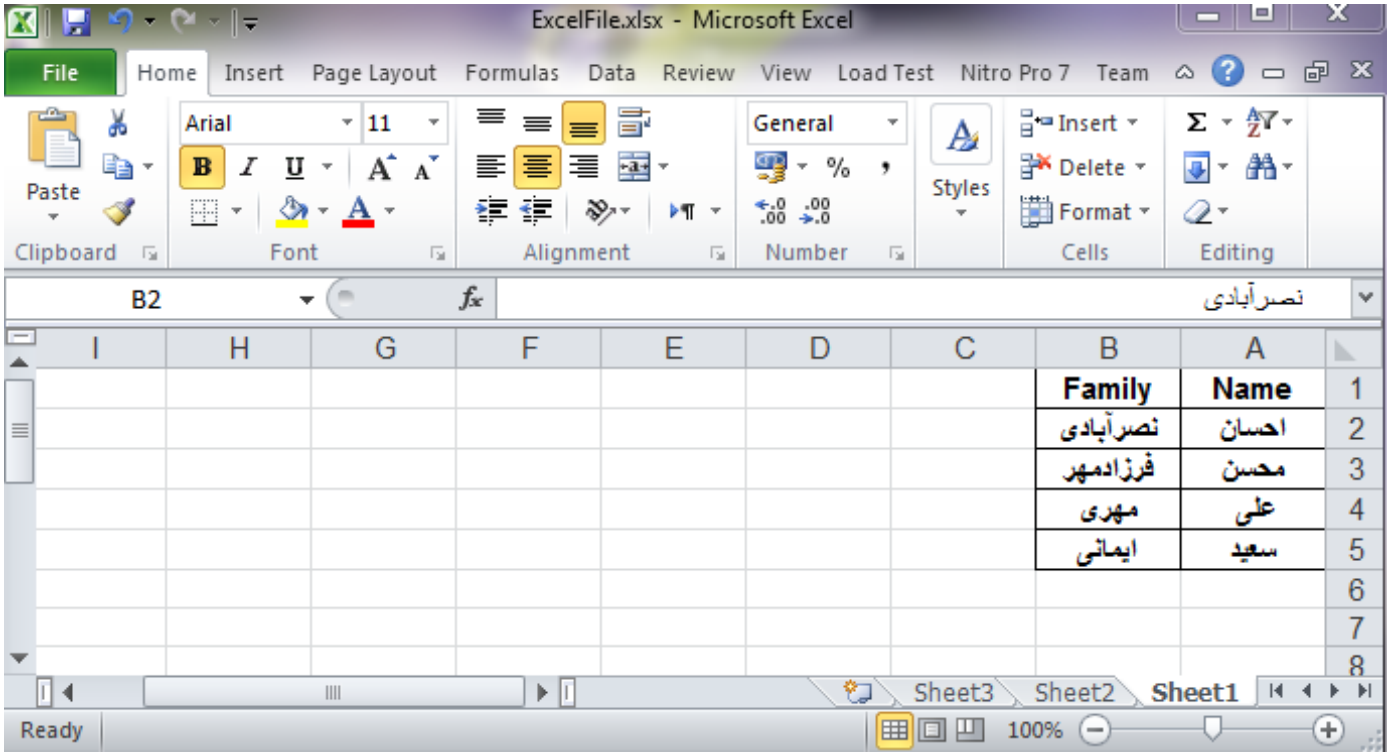
در این مقاله مروری سریع و کاربردی خواهیم داشت بر توانایی‌های مقدماتی LinqToExcel
در ابتدا می‌بایست **LinqToExcel** را از طریق NuGet به پروژه افزود.

```
PM> Install-Package LinqToExcel
```

و یا از طریق solution Explorer گزینه Manage NuGet Packages



اکنون فایل اکسل ذیل را در نظر بگیرید.



روش خواندن اطلاعات از فایل اکسل فوق تحت فرامین Linq و با مشخص کردن نام sheet مورد نظر توسط شئ ExcelQueryFactory بصورت زیر است.

```
string pathToExcelFile = @"C:\Users\MASOUD\Desktop\ExcelFile.xlsx";
var excel = new ExcelQueryFactory(pathToExcelFile);
            string sheetName = "Sheet1";
            var persons = from a in excel.Worksheet(sheetName) select a;
            foreach (var a in persons)
            {
                MessageBox.Show(a["Name"]+" "+a["Family"]);
            }
```

در صورتیکه بخواهیم انتقال اطلاعات فایل اکسل به جداول بانک اطلاعاتی مانند Sql Server بطور مثال با روش EF Entity Framework را انجام دهیم کلاس زیر با نام person را فرض نمایید.

```
public class Person
        {
            public string Name { get; set; }
            public string Family { get; set; }
        }
```

باید بدانید که بصورت پیشفرض سطر اول از فایل اکسل به عنوان نام ستون انتخاب می‌شود و می‌بایست جهت نگاشت با نام propertyهای کلاس ما دقیقاً همنام باشد.

```
string pathToExcelFile = @"C:\Users\MASOUD\Desktop\ExcelFile.xlsx";
            var excel = new ExcelQueryFactory(pathToExcelFile);
            string sheetName = "Sheet1";
            var persons = from a in excel.Worksheet<Person>(sheetName) select a;
            foreach (var a in persons)
            {
                MessageBox.Show(a.Name+" "+a.Family);
```

```
            }
```

اگر فایل اکسل ما ستون‌های بیشتری داشته باشد تنها ستونهای همنام با propertyهای کلاس ما به کلاس نگاشت پیدا می‌کند و سایر ستونها نادیده گرفته می‌شود.
در صورتیکه نام ستونهای فایل اکسل(سطر اول) با نام propertyهای کلاس یکسان نباشد جهت نگاشت آنها در کلاس می‌توان از متد AddMapping استفاده نمود.

```
string pathToExcelFile = @"C:\Users\MASOUD\Desktop\ExcelFile.xlsx";
            var excel = new ExcelQueryFactory(pathToExcelFile);
            string sheetName = "Sheet1";
            excel.AddMapping("Name","نام");
            excel.AddMapping("Family", "نام خانوادگی");
            var persons = from a in excel.Worksheet<Person>(sheetName) select a;
            foreach (var a in persons)
            {
                MessageBox.Show(a.Name+" "+a.Family);
            }
```

در کدهای بالا در صورتی که sheetName قید نشود بصورت پیشفرض Sheet1 از فایل اکسل انتخاب می‌شود.

```
var persons = from a in excel.Worksheet<Person>() select a;
```

همچنین می‌توان از اندیس جهت مشخص نمودن Sheet مورد نظر استفاده نمود که اندیس‌ها از صفر شروع می‌شوند.

```
var persons = from a in excel.Worksheet<Person>(0) select a;
```

توسط متد GetWorksheetNames می توان نام sheetها را بدست آورد.

```
public IEnumerable<string> getWorkSheets()
{
string pathToExcelFile = @"C:\Users\MASOUD\Desktop\ExcelFile.xlsx";

    var excel = new ExcelQueryFactory(pathToExcelFile);

    return excel.GetWorksheetNames();
}
```

و توسط متد GetColumnNames می توان نام ستونها را بدست آورد.

```
var SheetColumnNames = excel.GetColumnNames(sheetName);
```

همانطور که می‌بینید با روش توضیح داده شده در این مقاله به راحتی از فرامین Linq مانند where می‌توان در انتخاب اطلاعات از فایل اکسل استفاده نمود و سپس نتیجه را به جداول مورد نظر انتقال داد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۸/۰۶ ۲۲:۴۲

با تشکر از شما. این کتابخانه LinqToExcel کار کیست؟ سایت اصلی آن کجاست؟ مجوز استفاده از آن به چه صورتی است؟

نویسنده: مسعود حق شناس
تاریخ: ۱۳۹۲/۰۸/۰۶ ۲۳:۴۹

یک ویدیوی آموزشی کوتاه جالب با توضیحات راجع به این کتابخانه اینجا وجود داره که پیشنهاد میکنم ببینید. فقط من بدون فیلترشکن نتونستم صفحه اش و باز کنم.

نویسنده: مهدی
تاریخ: ۱۳۹۲/۰۸/۰۷ ۱۰:۱۹

سلام
اگر نام ستون‌ها رو نداشته باشیم و فقط بخواهیم اطلاعات داخل شیت رو بدست بیاریم چیکار باید کرد در ضمن آیا باز نیازی به اسمبلی Microsoft.Office.Interop.Excel هست یا نه ؟

نویسنده: مسعود حق شناس
تاریخ: ۱۳۹۲/۰۸/۰۷ ۲۲:۳۹

با استفاده از متد WorksheetNoHeader و با وارد کردن شماره اندیس می‌توانید به اطلاعات سلولها دست پیدا کنید. در مثال زیر تمام سطرهایی که ستون دوم آنها شهر مشهد است انتخاب می‌شوند و سپس سلولهای آن سطرها نمایش داده می‌شوند.

```
var excel = new ExcelQueryFactory(pathToExcelFile);
            string sheetName = "Sheet1";
            var persons = from a in excel.WorksheetNoHeader(sheetName)
                          where a[1] == "Mashhad" //مقدار در ستون دوم
                          select a;
            foreach (var a in persons)
              {
                  for(int i=0;i<a.Count;i++)
                        MessageBox.Show(a[i]);
              }
```

به اسمبلی Microsoft.Office.Interop.Excel نیازی نیست و
LinqToExcel.dll و Remotion.Data.Linq.dll مورد نیاز است.

نویسنده: بهراد
تاریخ: ۱۳۹۲/۰۸/۰۸ ۸:۴۹

سلام من این خطا رو میگیرم

```
The 'Microsoft.ACE.OLEDB.12.0' provider is not registered on the local machine.
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۸ ۱۰:۴۶

مطابق توضیحات آن ، نیاز به AccessDatabaseEngine نیز دارد (Microsoft.ACE.OLEDB.12.0 مربوط به اکسس 2010 است).
احتمالا برای سازگاری با نگارش‌های قدیمی اکسل که با فرمت OpenXML نیستند از این نوع رشته اتصالی مخصوص اکسس 2010 در پشت صحنه استفاده کرده:

```
Driver={Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)};DBQ=path to xls/xlsx/xlsm/xlsb file
```

نویسنده: noth50
تاریخ: ۱۳۹۲/۰۹/۱۱ ۱۰:۴۴

بادرود
ممنون از اطلاعات مفید شما .
من در یک پروژه تحت وب از طریق فایل اکسل اطلاعات را از کاربر دریافت می‌کنم و نیاز دارم این اطلاعات را در داخل دیتابیس ذخیره کنم از آنجا که تعداد رکوردهای فایل اکسل نامشخص است به چه صورت باید این کد را بنویسم
ممنون

عنوان: **ذخیره سریع ده‌ها هزار رکورد دیتاتیبل در اکسل**
نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۹:۳۵
آدرس: www.dotnettips.info
گروه‌ها: C#, Excel

یکی از امکاناتی که در نرم افزارهای اتوماسیون مورد نیاز است، ذخیره اطلاعات، داخل فایل اکسل است و در صورتی که حجم این اطلاعات زیاد باشد زمان زیادی صرف این عمل خواهد شد. در زیر کلاسی را برای شما آماده نموده‌ام که 20 هزار رکورد را در 4 ثانیه، در فایل اکسل ذخیره می‌نماید. در این روش با استفاده از یک آرایه به نام rawdata این عمل انجام شده.
توضیحات کدها نیز به صورت comment در کنار کدها آورده شده است.

```
//using System;
    //using System.Data;
    //using Microsoft.Office.Interop.Excel;

    class FastExportingMethod
    {
        //System.Data.DataTable dt= دیتاتیبل که حاوی اطلاعات می‌باشد
        //outputPath= مسیر ذخیره شدن
        public static string ExportToExcel(System.Data.DataTable dt, string outputPath)
        {
            try
            {
                // ساخت یک شی اکسل
                ApplicationClass excelApp = new ApplicationClass();

                // ساخت یک WorkBook جدید
                Workbook excelWorkbook = excelApp.Workbooks.Add(Type.Missing);

                int sheetIndex = 0;

                // ساخت آرایه به طول تعداد سطرهای دیتاتیبل+1 و تعداد ستونهای دیتاتیبل
                object[,] rawData = new object[dt.Rows.Count + 1, dt.Columns.Count];

                // کپی نام ستون‌های دیتاتیبل به عنوان هدر برای فایل اکسل در اولین سطر از آرایه
                for (int col = 0; col < dt.Columns.Count; col++)
                {
                    rawData[0, col] = dt.Columns[col].ColumnName;
                }

                // کپی اطلاعات دیتاتیبل به داخل آرایه
                for (int col = 0; col < dt.Columns.Count; col++)
                {
                    for (int row = 0; row < dt.Rows.Count; row++)
                    {
                        rawData[row + 1, col] = dt.Rows[row].ItemArray[col].ToString();
                    }
                }

                // محاسبه نام ستونهای اکسل

                string finalColLetter = string.Empty;
                string colCharset = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
                int colCharsetLen = colCharset.Length;

                if (dt.Columns.Count > colCharsetLen)
                {
                    finalColLetter = colCharset.Substring((dt.Columns.Count - 1) / colCharsetLen - 1,
1);
                }

                finalColLetter += colCharset.Substring((dt.Columns.Count - 1) % colCharsetLen, 1);

                // ساخت یک Sheet
                Worksheet excelSheet = (Worksheet)excelWorkbook.Sheets.Add(
                    excelWorkbook.Sheets.get_Item(++sheetIndex),
                    Type.Missing, 1, XlSheetType.xlWorksheet);
                //تنظیم نام شیت به نام دلخواه
                excelSheet.Name = "List";
                //تنظیم خاصیت راست به چپ برای نمایش اطلاعات
                excelSheet.DisplayRightToLeft = true;


                // تعیین محدوده سطرها و ستونها
                string excelRange = string.Format("A1:{0}{1}",finalColLetter, dt.Rows.Count + 1);
                //انتقال اطلاعات از آرایه به شیت مورد نظر
```

```
                excelSheet.get_Range(excelRange, Type.Missing).Value2 = rawData;

                // ضخیم کردن اولین سطر برای عنوان ستونها
                ((Range)excelSheet.Rows[1, Type.Missing]).Font.Bold = true;

                // تنظیم عرض ستونها به اندازه محتوای ستونها
                for (int col = 0; col < dt.Columns.Count; col++)
                {
                    ((Range)excelSheet.Columns[col + 1]).EntireColumn.AutoFit();
                }


                //ذخیره و بستن  Workbook
                excelWorkbook.SaveAs(outputPath, XlFileFormat.xlWorkbookNormal, Type.Missing,
                    Type.Missing, Type.Missing, Type.Missing, XlSaveAsAccessMode.xlExclusive,
                    Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);
                excelWorkbook.Close(true, Type.Missing, Type.Missing);
                excelWorkbook = null;

                excelApp.Quit();
                excelApp = null;

                // Collect the unreferenced objects
                GC.Collect();
                GC.WaitForPendingFinalizers();

                return "اطلاعات شما در مسیر انتخاب شده ذخیره گردید";
            }
            catch (Exception ex)
            {
                //بدست آوردن کد خطا برای مدیریت خطاها
                int code = System.Runtime.InteropServices.Marshal.GetExceptionCode();

                return ex.Message + code;
            }
        }
    }
```

نظرات خوانندگان

نویسنده: راضیه
تاریخ: ۱۳۹۲/۱۱/۱۷ ۱۹:۵۶

کتابخانه NPOI نیز دارای سرعت بسیار بالایی است.پیشنهاد می‌کنم حتما امتحانش کنید

http://npoi.codeplex.com/

نمونه: http://www.zachhunter.com/2010/05/getting-started-with-npoi/

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۷ ۲۰:۲

برای کار با اکسل، اگر 2007 به بعد مدنظر است، بدون نیاز به نصب اکسل می‌شود با EPPlus هم کار کرد.

عنوان: **خواندن سریع اطلاعات فایل اکسل و ذخیره در بانک SQL**
نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۱/۱۷ ۱۸:۳۵
آدرس: www.dotnettips.info
گروه‌ها: C#, SQL Server, Excel

یکی از زمانبرترین عملیاتها در نرم افزارهای اتوماسیون، خواندن اطلاعات از فایل‌های اکسل با حجم بالا است. در صورتی که این کار را می‌توان با استفاده از کلاس SqlBulkCopy به سرعت انجام داد. در ادامه نحوه استفاده از این کلاس، همراه نمونه کدها آورده شده است.
توضیحات به صورت Comment است.

```
try
            {
//انتخاب فایل اکسل
                OpenFileDialog ofd = new OpenFileDialog();
                ofd.Title = "انتخاب فایل اکسل حاوی اطلاعات";
                ofd.Filter = "2003 فایل اکسل (*.xls)|*.xls|فایل اکسل 2007 به بعد(*.xlsx)|*.xlsx";
                if (ofd.ShowDialog() == DialogResult.OK)
                {
                    string excelConnectionString = "";
                    string SourceFilePath = ofd.FileName;
//ایجاد کانکشن استرینگ برا خواندن کل اطلاعات از فایل اکسل و ریختن آنها در یک دیتاتیبل به نام dt
                    if (System.IO.Path.GetExtension(ofd.FileName) == ".xlsx")
//تشخیص نوع فایل اکسل برای ایجاد کانکشن استرینگ برای نسخه‌های مختلف اکسل
                        excelConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
SourceFilePath + ";Extended Properties=Excel 12.0";
                    else if (System.IO.Path.GetExtension(ofd.FileName) == ".xls")
                        excelConnectionString = "Provider=Microsoft.Jet.Oledb.4.0;Data Source=" +
SourceFilePath + ";Extended Properties=Excel 8.0";

                    DataTable dt = new DataTable("tblinfos");


                    using (System.Data.OleDb.OleDbConnection connection = new
System.Data.OleDb.OleDbConnection(excelConnectionString))
                    {

                        connection.Open();

                        System.Data.OleDb.OleDbDataAdapter da = new
System.Data.OleDb.OleDbDataAdapter("SELECT * FROM [List$]", connection);//List نام شیت در فایل اکسل
است


                        da.Fill(dt);
                        connection.Close();

                    }

                    using (System.Data.OleDb.OleDbConnection connection = new
System.Data.OleDb.OleDbConnection(excelConnectionString))
                    {
                        this.Cursor = Cursors.WaitCursor;

                        connection.Open();
//ایجاد ارتباط با بانک اس کیو ال
                        string sqlConnectionString = @"Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\BankDPR.mdf;Max Pool Size=6000; Connection
Timeout=50;Integrated Security=True;User Instance=True";


                        Dataaccess db = new Dataaccess();
                        DataTable dtr = db.select("Select Top(1) * From tblinfos");//بدست آورن نام
ستون‌های جدول مورد نظر برای تطبیق با ستونهای فایل اکسل
                        //*******************   Fast Copy
                        using (System.Data.SqlClient.SqlBulkCopy bulkCopy = new
System.Data.SqlClient.SqlBulkCopy(sqlConnectionString,
System.Data.SqlClient.SqlBulkCopyOptions.KeepIdentity))//تعریف یک شی از کلاس SqlBulkCopy
                        {
                            for (int i = 1; i < dtr.Columns.Count; i++)
                            {
                                bulkCopy.ColumnMappings.Add(dt.Columns[i - 1].Caption,
dtr.Columns[i].Caption);//نام ستونهای فایل اکسل با نام ستونهای جدول ColumnMappings با استفاده از خاصیت
تطبیق داده می‌شود sql مورد نظر بانک
                            }

                            bulkCopy.DestinationTableName = "tblinfos";//مشخص نمودن نام جدول که قرار
```

```
است اطلاعات درون ان کپی گردد
                        bulkCopy.WriteToServer(dt);//نام با دیتاتیبل از اطلاعات کپی عملیات انجام dt به
بانک

                    }

                    this.Cursor = Cursors.Arrow;
                    MessageBox.Show("اطلاعات از فایل شما خوانده شد");

                }
            }
        }
        catch (Exception ex)
        {
            this.Cursor = Cursors.Arrow;

            MessageBox.Show(ex.Message, "error");
        }
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۷ ۲۱:۲۲

احتمالا این قطعه کد مستقیما از داخل سورس یکی از پروژه‌های شما بیرون آمده (مثال نیست؛ واقعی هست). می‌شد کمی اون رو refactor کرد مثلا یک متد از داخلش بیرون آورد که این متد داخلش مسیج باکس نباشه یا باز کردن یک صفحه دیالوگ و تغییر کرسر. try و catch هم نداشته باشه چون باید در یک سطح بالاتر catch بشه مشکلاتش. اون selectها مثلا می‌شدند چند پارامتر، برای اینکه این کد قابلیت استفاده مجدد بهتری پیدا کنه. یا مثلا اون sqlConnectionString از داخل کدها بیرون می‌اومد و می‌شد یک پارامتر جدید. نمایش کرسر هم داخل این متد قرار نمی‌گرفت. نام جدول نهایی هم مثلا یک پارامتر دیگر می‌شد برای سهولت استفاده مجدد و همچنین تست بهتر یک قطعه کوچک از کار. خود متد اصلی هم می‌شد دو متد کوچک‌تر؛ یکی کار load رو انجام می‌داد و دیگری کار insert سریع.

نویسنده: پالادین
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۳:۳۸

من هم با دوستمون محسن خان موافقم. اتفاق‌ها زیادی در مورد این کد باید بیافته تا به یک کد خوب تبدیل بشه. شما کاملا به صورت Smart UI کد زدید. امیدوارم وقت بزارید و این کد رو بهینه کنید. توی همین سایت مثال‌های خوبی واسه‌ی یادگیری هست..

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۱:۳۶

بله این کد رو من دقیقا از یکی از برنامه هام آوردم و مال چندین سال پیشه.
اما خوبیش اینه که بدون خطا جواب میده و فقط کافیه اسم جداول رو عوض کنید.

نویسنده: محمدی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۲۹

با سلام و تشکر. این برنامه پیغام میدهد List وجود ندارد.

```
SELECT * FROM [List$]
```

لطفا راهنمایی بفرمایید من چطور میتوانم نام شیت‌ها را بدست بیاورم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۴۷

در کد فوق اکثر قسمت‌ها بر اساس یک سری پیش فرض مشخص تهیه شده‌اند و آن‌ها را تبدیل به پارامتر متغیر نکرده‌اند. نام شیت، همان نام برگه جاری است:

نویسنده: احمد زواری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۲۳:۴۰

سلام. من از این روش برای خواندن فایل موجود در آدرس
http://www.site.com/market/export.aspx?type=dtod&date=20120405 استفاده کردم، ولی متاسفانه فیلدهای عددی را NULL مقدار میگیرد. مشکل از چیست و چه جوری برطرف میشود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۲ ۰:۲۴

این فایل استاندارد اکسل نیست. خروجی آن را با نوت پد باز کنید؛ یک فایل HTML معمولی است.

نویسنده: احمد زواری
تاریخ: ۱۳۹۳/۰۴/۱۲ ۳:۳۰

الان من نیاز شدید دارم این فایل رو تبدیل به SQL کنم، و چون هر روز این فایل تغییر میکنه نمیتونم توی فایل تغییری بدم، چه راه حلی هست برای این کار؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۲ ۹:۱۹

فایل‌های HTML را با استفاده از کتابخانه‌ی HTML Agility Pack پردازش می‌کنند.