عنوان: **نحوه ایجاد یک گزارش فاکتور فروش توسط PdfReport**
نویسنده: وحید نصیری
تاریخ: ۲۱:۵۹ ۱۳۹۱/۰۷/۲۳
آدرس: www.dotnettips.info
گروه‌ها: PdfReport

شکل زیر را که شبیه به یک فاکتور فروش است درنظر بگیرید:

گزارش جدید ما

ردیف	شماره	نام	نام خانوادگی	موجودی
۱	۰	نام ۰	نام خانوادگی ۰	۱,۰۰۰
۲	۱	نام ۱	نام خانوادگی ۱	۱,۰۰۱
۳	۲	نام ۲	نام خانوادگی ۲	۱,۰۰۲
۴	۳	نام ۳	نام خانوادگی ۳	۱,۰۰۳
۵	۴	نام ۴	نام خانوادگی ۴	۱,۰۰۴
۶	۵	نام ۵	نام خانوادگی ۵	۱,۰۰۵
۷	۶	نام ۶	نام خانوادگی ۶	۱,۰۰۶
			جمع صفحه	۷,۰۲۱
			جمع کل	۷,۰۲۱
			مالیات	۱۵۴
			عوارض	۱۲۶
			جمع کل	۷,۳۰۱
			قابل پرداخت	هفت هزار و سیصد و یک ریال

نکته‌ای که در اینجا مدنظر است، دسترسی به عدد جمع آخر گزارش و سپس بر اساس آن، ساخت دو ستون اضافی ذیل جدول اصلی گزارش است که موارد مالیات، عوارض، جمع کل و مبلغ به حروف را نسبت به مثال‌های قبلی، اضافه‌تر دارد.

در ادامه کدهای کامل این مثال را مشاهده می‌کنید. همچنین این کد و کلاس‌های وابسته به آن مانند User و TransparentTemplate به سورس‌های کتابخانه PdfReport نیز اضافه شده‌اند .

```
using System;
using System.Collections.Generic;
using iTextSharp.text;
using iTextSharp.text.pdf;
using PdfReportSamples.Models;
using PdfReportSamples.Templates;
using PdfRpt.Core.Contracts;
using PdfRpt.Core.Helper;
using PdfRpt.FluentInterface;

namespace PdfReportSamples.Tax
{
    public class TaxPdfReport
    {
        public IPdfReportData CreatePdfReport()
        {
            return new PdfReport().DocumentPreferences(doc =>
            {
                doc.RunDirection(PdfRunDirection.RightToLeft);
                doc.Orientation(PageOrientation.Portrait);
                doc.PageSize(PdfPageSize.A4);
```

```
                doc.DocumentMetadata(new DocumentMetadata { Author = "Vahid", Application = "PdfRpt",
Keywords = "Test", Subject = "Test Rpt", Title = "Test" });
            })
            .DefaultFonts(fonts =>
            {
                fonts.Path(AppPath.ApplicationPath + "\\fonts\\irsans.ttf",
                                Environment.GetEnvironmentVariable("SystemRoot") +
"\\fonts\\verdana.ttf");
            })
            .PagesFooter(footer =>
            {
                footer.DefaultFooter(DateTime.Now.ToString("MM/dd/yyyy"));
            })
            .PagesHeader(header =>
            {
                header.DefaultHeader(defaultHeader =>
                {
                    defaultHeader.ImagePath(AppPath.ApplicationPath + "\\Images\\01.png");
                    defaultHeader.Message("گزارش جدید ما");
                });
            })
            .MainTableTemplate(template =>
            {
                template.CustomTemplate(new TransparentTemplate());
            })
            .MainTablePreferences(table =>
            {
                table.ColumnsWidthsType(TableColumnWidthType.Relative);
            })
            .MainTableDataSource(dataSource =>
            {
                var listOfRows = new List<User>();
                for (int i = 0; i < 7; i++)
                {
                    listOfRows.Add(new User { Id = i, LastName = "نام خانوادگی " + i, Name = "نام " + i,
Balance = i + 1000 });
                }
                dataSource.StronglyTypedList<User>(listOfRows);
            })
            .MainTableSummarySettings(summarySettings =>
            {
                summarySettings.OverallSummarySettings("جمع کل");
                summarySettings.PreviousPageSummarySettings("نقل از صفحه قبل");
                summarySettings.PageSummarySettings("جمع صفحه");
            })
            .MainTableColumns(columns =>
            {
                columns.AddColumn(column =>
                {
                    column.PropertyName("rowNo");
                    column.IsRowNumber(true);
                    column.CellsHorizontalAlignment(HorizontalAlignment.Center);
                    column.IsVisible(true);
                    column.Order(0);
                    column.Width(1);
                    column.HeaderCell("ردیف");
                });

                columns.AddColumn(column =>
                {
                    column.PropertyName<User>(x => x.Id);
                    column.CellsHorizontalAlignment(HorizontalAlignment.Center);
                    column.IsVisible(true);
                    column.Order(1);
                    column.Width(2);
                    column.HeaderCell("شماره");
                });

                columns.AddColumn(column =>
                {
                    column.PropertyName<User>(x => x.Name);
                    column.CellsHorizontalAlignment(HorizontalAlignment.Center);
                    column.IsVisible(true);
                    column.Order(2);
                    column.Width(3);
                    column.HeaderCell("نام");
                });

                columns.AddColumn(column =>
                {
                    column.PropertyName<User>(x => x.LastName);
```

```
                    column.CellsHorizontalAlignment(HorizontalAlignment.Center);
                    column.IsVisible(true);
                    column.Order(3);
                    column.Width(3);
                    column.HeaderCell("نام خانوادگی");
                });

                columns.AddColumn(column =>
                {
                    column.PropertyName<User>(x => x.Balance);
                    column.CellsHorizontalAlignment(HorizontalAlignment.Center);
                    column.IsVisible(true);
                    column.Order(4);
                    column.Width(2);
                    column.HeaderCell("موجودی");
                    column.ColumnItemsTemplate(template =>
                    {
                        template.TextBlock();
                        template.DisplayFormatFormula(obj => obj == null ? string.Empty :
string.Format("{0:n0}", obj));
                    });
                    column.AggregateFunction(aggregateFunction =>
                    {
                        aggregateFunction.NumericAggregateFunction(AggregateFunction.Sum);
                        aggregateFunction.DisplayFormatFormula(obj => obj == null ? string.Empty :
string.Format("{0:n0}", obj));
                    });
                });

            })
            .MainTableEvents(events =>
            {
                events.DataSourceIsEmpty(message: "There is no data available to display.");

                events.MainTableAdded(args =>
                {
                    var balanceData = args.LastOverallAggregateValueOf<User>(u => u.Balance);
                    var balance = double.Parse(balanceData,
System.Globalization.NumberStyles.AllowThousands);

                    var others = Math.Round(balance * 1.8 / 100);
                    var tax = Math.Round(balance * 2.2 / 100);
                    var total = balance + tax + others;

                    var taxTable = new PdfPTable(args.Table); // Create a clone of the MainTable's
structure

                    taxTable.AddSimpleRow(
                        null /* null = empty cell */, null, null,
                        (data, cellProperties) =>
                        {
                            data.Value = "مالیات";
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.HorizontalAlignment = HorizontalAlignment.Left;
                        },
                        (data, cellProperties) =>
                        {
                            data.Value = string.Format("{0:n0}", tax);
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.BorderColor = BaseColor.LIGHT_GRAY;
                            cellProperties.ShowBorder = true;
                        });

                    taxTable.AddSimpleRow(
                        null, null, null,
                        (data, cellProperties) =>
                        {
                            data.Value = "عوارض";
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.HorizontalAlignment = HorizontalAlignment.Left;
                        },
                        (data, cellProperties) =>
                        {
                            data.Value = string.Format("{0:n0}", others);
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.BorderColor = BaseColor.LIGHT_GRAY;
                            cellProperties.ShowBorder = true;
                        });

                    taxTable.AddSimpleRow(
                        null, null, null,
```

```
                        (data, cellProperties) =>
                        {
                            data.Value = "جمع کل";
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.HorizontalAlignment = HorizontalAlignment.Left;
                        },
                        (data, cellProperties) =>
                        {
                            data.Value = string.Format("{0:n0}", total);
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.BorderColor = BaseColor.LIGHT_GRAY;
                            cellProperties.ShowBorder = true;
                        });

                    taxTable.AddSimpleRow(
                        null, null, null,
                        (data, cellProperties) =>
                        {
                            data.Value = "قابل پرداخت";
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.HorizontalAlignment = HorizontalAlignment.Left;
                        },
                        (data, cellProperties) =>
                        {
                            data.Value = total.NumberToText(Language.Persian) + " ریال";
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.BorderColor = BaseColor.LIGHT_GRAY;
                            cellProperties.ShowBorder = true;
                            cellProperties.PdfFontStyle = DocumentFontStyle.Bold;
                        });

                    args.PdfDoc.Add(taxTable);
                });
            })
            .Export(export =>
            {
                export.ToExcel();
            })
            .Generate(data => data.AsPdfFile(AppPath.ApplicationPath + "\\Pdf\\TaxReportSample.pdf"));
        }
    }
}
```

توضیحات:

تنها تفاوت این مثال با مثال‌های قبلی، کدهای مرتبط با متد events.MainTableAdded می‌باشند.

توسط متد args.LastOverallAggregateValueOf، می‌توان به مقدار نهایی متد تجمعی تعریف شده برای یک ستون خاص دسترسی یافت:

```
var balanceData = args.LastOverallAggregateValueOf<User>(u => u.Balance);
var balance = double.Parse(balanceData, System.Globalization.NumberStyles.AllowThousands);
```

سپس بر این اساس، امکان محاسبه مالیات و عوارض میسر می‌شود:

```
var others = Math.Round(balance * 1.8 / 100);
var tax = Math.Round(balance * 2.2 / 100);
var total = balance + tax + others;
```

در ادامه نیاز داریم تا یک جدول جدید را ذیل جدول اصلی ایجاد کنیم. نکته مهم این جدول جدید، هماهنگی عرض ستون‌های آن با ستون‌های جدول اصلی است. به همین منظور می‌توان از خاصیت args.Table جهت دسترسی به خواص جدول اصلی استفاده کرد و جدول جدیدی را ایجاد نمود:

```
var taxTable = new PdfPTable(args.Table);
```

از اینجا به بعد دیگر به عهده خودتان است. می‌توانید از دانش iTextSharp خود استفاده کرده و ردیف‌های این جدول جدید را پر کنید. یا اینکه می‌توانید از متد کمکی توکار AddSimpleRow به نحو زیر استفاده نمائید:

```
taxTable.AddSimpleRow(
```

```
                        null /* null = empty cell */, null, null,
                        (data, cellProperties) =>
                        {
                            data.Value = "مالیات";
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.HorizontalAlignment = HorizontalAlignment.Left;
                        },
                        (data, cellProperties) =>
                        {
                            data.Value = string.Format("{0:n0}", tax);
                            cellProperties.PdfFont = args.PdfFont;
                            cellProperties.BorderColor = BaseColor.LIGHT_GRAY;
                            cellProperties.ShowBorder = true;
                        });
```

با توجه به اینکه قصد نداریم در سه ستون اول این جدول جدید، عنصری را نمایش دهیم، آن‌ها را با null مقدار دهی کرده و سپس ستون برچسب و ستون مقدار را اضافه می‌کنیم (آرگومان‌های این متد به صورت params تعریف شده‌اند. بنابراین هر تعداد ستون که نیاز باشد قابل تعریف است).

با مقدار دهی data، مقدار مورد نظر در آن سلول ثبت می‌گردد. با مقدار دهی خواص cellProperties، نوع قلم، جهت قرارگیری و سایر تنظیماتی را که ملاحظه می‌کنید، می‌توان اعمال کرد.

و در آخر لازم است که این جدول جدید را به شیء Document اضافه کنیم تا نمایش داده شود:

```
args.PdfDoc.Add(taxTable);
```

یک نکته:

متد NumberToText جزئی از کتابخانه PdfReport (تعریف شده در فضای نام PdfRpt.Core.Helper) است و برای نمایش رقم به حروف می‌تواند مورد استفاده قرار گیرد:

```
total.NumberToText(Language.Persian)
```

نظرات خوانندگان

نویسنده: mrm
تاریخ: ۱۳۹۱/۰۷/۳۰ ۲۲:۴۷

آقای نصیری با سلام
میخواستم ببینم آیا میتونم از pdfreport برای تولید برگه امتحان تستی استخراج شده از بانک سوال استفاده کنم و آیا شما توصیه میکنید یا نه؟
برگه شامل یک سربرگ و حاشیه هم باید باشه و سوالات تستی با گزینه هاشون توی اون باید به شکل مرتبی برای چاپ حاضر بشن
خیلی ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۳۰ ۲۳:۱۰

سلام؛
بله. ولی فکر نمی‌کنم این بله به درد شما بخورد. به همین جهت در این قسمت مخصوص سایت، شمای بانک (ساختار بانک)، به علاوه شکل نهایی مدنظر را به صورت یک بازخورد جدید ارسال کنید. من کدهای گزارش رو براتون به شکل یک مثال پیوست می‌کنم.

نویسنده: م دانا
تاریخ: ۱۳۹۱/۱۱/۱۴ ۱۸:۲۲

با سلام و تشکر از زحمات شما در ارتباط با کتابخانه PDFReport.
متاسفانه متد AddSimpleRow را در مثال فوق نمی‌شناسد! چه دلیلی می‌تواند داشته باشد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۴ ۱۸:۳۹

آخرین تغییرات این مثال رو در اینجا می‌تونید مشاهده کنید. فقط در در آن PdfPTable تبدیل به PdfGrid شده.

نویسنده: پویا
تاریخ: ۱۳۹۲/۰۲/۲۴ ۱۵:۳۹

با عرض سلام.

فرض می‌کنیم که هدر ما از سه قسمت هدر و جدول (همان جدولی که برای فاکتور در اینجا در نظر گرفته شده است)و فوتر تشکیل شده است حال چگونه می‌توانم فونت جدول را تعیین کنم یعنی این که سایز اون را تغییر دهم یا اینکه خود فونت را تغییر دهم. ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۴ ۱۶:۵۲

- فونتی که در قسمت DefaultFonts تعریف می‌شود (که در اینجا متدهای تنظیم اندازه و رنگ نیز وجود دارند) در حقیقت فونت اصلی جدول گزارش است. اگر هدر و فوتر، فونتی رو معرفی نکنه، این فونت برای آن‌ها هم استفاده خواهد شد (در حالت‌های DefaultFooter و DefaultHeader). اما می‌شود هدر و فوتر سفارشی هم تعریف کرد؛ با هر نوع طراحی و هر نوع فونت دلخواهی که صلاح می‌دونید (مراجعه کنید به مثال InlineProviders مجموعه مثال‌ها ، که در آن خاصیت PdfFont مستقل هم وجود دارد).
- به علاوه در اینجا حتی می‌شود بنابر شرایط و مقادیر سلول‌ها، فونت و رنگ خاصی را به مقادیر یک سلول اعمال کرد. نمونه‌اش

در مثال CustomCellTemplate وجود دارد.
- در این حالت‌های خاص باید IPdfFont را پیاده سازی کنید. یک نمونه پروایدر ساده ساز در اینجا به نام کلاس GenericFontProvider برای اینکار تدارک دیده شده. مثالی در این مورد: InjectCustomRows جهت مقدار دهی CellBasicProperties.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۴/۰۲/۲۸ ۱۰:۵

سلام، آیا می‌توان از این کتابخانه خروجی اکسل گرفت؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۲/۲۸ ۱۰:۴۹

خروجی اکسل این کتابخانه، داخل خود فایل PDF قرار داده می‌شود:

```
.Export(export =>
            {
                export.ToExcel();
            })
```