

# Web-aplikacije

## Izrada izvješća (PDF)

2019/20.15

# Alat za izradu izvješća

2

➤ Za izradu izvješća u PDF formatu koristit će se alat PdfReport.Core

➤ <https://github.com/VahidN/PdfReport.Core>



**PdfRpt.Core** by Vahid Nasiri

📄 v1.1.4

PdfReport.Core is a code first reporting engine, which is built on top of the iTextSharp.LGPLv2.Core and EPPlus.Core libraries.

➤ Primjeri sadrže 3 vrsta izvješća:

➤ Jednostavni tablični izvještaj

➤ Popis svih država

➤ Tablični izvještaj sa slikama i sumom vrijednosti nekog stupca

➤ Prikaz 10 najskupljih artikala sa slikom

➤ Master-detail primjer koji demonstrira grupiranje elemenata

➤ N najboljih kupnji (tj. dokumenata s najvećom vrijednosti zajedno s popisom stavki)

# Inicijalne postavke za izvješća (1)

3

- Razred za rad s izvješćima je *PdfReport* koji sadrži postupke koji vraćaju referencu na vlastiti objekt, pa se postupci mogu vezati
  - Inicijalno se postavljaju metapodaci (autori, naslov, ...) i orijentacija
  - Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
private PdfReport CreateReport(string naslov) {  
    var pdf = new PdfReport();  
    pdf.DocumentPreferences(doc =>  
    {  
        doc.Orientation(PageOrientation.Portrait);  
        doc.PageSize(PdfPageSize.A4);  
        doc.DocumentMetadata(new DocumentMetadata {  
            Author = "FER-ZPR", Application = "Firma.MVC Core",  
            Title = naslov  
        });  
        doc.Compression(new CompressionSettings {  
            EnableCompression = true,  
            EnableFullCompression = true  
        });  
    });  
}
```

## Inicijalne postavke za izvješća (2)

4

➡ Postavlja se vrsta predloška, automatska širina stupaca, može se ograničiti broj redaka po stranici, broj grupa po stranici i slično.

➡ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
private PdfReport CreateReport(string naslov) {  
    var pdf = new PdfReport();  
    ...  
    .MainTableTemplate(template => {  
        template.BasicTemplate(BasicTemplate.ProfessionalTemplate);  
    })  
    .MainTablePreferences(table => {  
        table.ColumnsWidthType(TableColumnWidthType.Relative);  
        //table.NumberOfDataRowsPerPage(20);  
        table.GroupsPreferences(new GroupsPreferences {  
            GroupType = GroupType.HideGroupingColumns,  
            RepeatHeaderRowPerGroup = true,  
            ShowOneGroupPerPage = true  
        });  
        table.SpacingAfter(4f);  
    });  
}
```

# Zaglavlje i podnožje jednostavnih izvješća

5

➤ Jednostavna izvješća će u zaglavlju imati naslov izvješća, a u podnožju trenutni datum

➤ Potrebno postaviti smjer teksta na LeftToRight

➤ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Drzave() {  
    ...  
    PdfReport report = CreateReport(naslov);  
    report.PagesFooter/footer => {  
        footer.DefaultFooter(DateTime.Now.ToString("dd.MM.yyyy."));  
    })  
    .PagesHeader/header => {  
        header.CacheHeader(cache: true);  
        header.DefaultHeader(defaultHeader => {  
            defaultHeader.RunDirection(  
                PdfRunDirection.LeftToRight);  
            defaultHeader.Message(naslov);  
        });  
    });  
};
```

# Izvor podataka za izvješće

6

➤ U svim primjerima kao izvor podataka služit će lista čiji su elementi objekti nekog prezentacijskog modela ili anonimnog razreda

➤ Podaci se u izvještaj dodaju redom kojim se nalaze u listi

➤ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Drzave() {  
    var drzave = await ctx.Drzava  
        .AsNoTracking()  
        .OrderBy(d => d.NazDrzave)  
        .ToListAsync();  
}
```

```
public async Task<IActionResult> Artikli() {  
    var artikli = await ctx.Artikl  
        .AsNoTracking()  
        .Where(a => a.SlikaArtikla != null)  
        .OrderByDescending(a => a.CijArtikla)  
        .Select(a => new {  
            a.SifArtikla, a.NazArtikla,  
            a.CijArtikla, a.SlikaArtikla  
        })  
        .Take(10).ToListAsync();  
}
```

# Postavljanje izvora podataka

7

- Izvor se postavlja pozivom postupka *MainTableDataSource* i pisanjem odgovarajućeg delegata tipa *Action<MainTableDataSourceBuilder>*
  - Nad podatkom tipa *MainTableDataSourceBuilder* poziva se postupak *StronglyTypedList* te se kao argument predaje pripremljena lista
    - Alternative: postupci *SqlDataReader*, *CustomDataSource*, ...
- Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Drzave() {  
    var drzave = await ctx.Drzava  
        .AsNoTracking()  
        .OrderBy(d => d.NazDrzave)  
        .ToListAsync();  
  
    report.MainTableDataSource(dataSource =>  
        dataSource.StronglyTypedList(drzave));  
}
```

# Definiranje stupaca

8

➡ Željeni stupci se navode u postupku *MainTableColumns*

➡ Stupcima se postavlja poredak (ako se izostavi koristi se redoslijed navođenja) relativna širina, vidljivost, zaglavlje te naziv svojstva

➡ Moguće imati redak s rednim brojem

➡ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs (Drzave)

```
report.MainTableColumns(columns => {  
    columns.AddColumn(column => {  
        column.IsRowNumber(true);  
        column.CellsHorizontalAlignment(HorizontalAlignment.Right);  
        column.IsVisible(true); column.Order(0); column.Width(1);  
        column.HeaderCell("#",  
            horizontalAlignment: HorizontalAlignment.Right);  
    });  
    columns.AddColumn(column => {  
        column.PropertyName(nameof(Drzava.OznDrzave));  
        column.Order(1); column.Width(2);  
        column.HeaderCell("Oznaka države");  
    });  
});
```



# Preuzimanje izvješća

9

➤ Binarni sadržaj izrađenog izvješća vraća se korisniku korištenjem postupka File (naslijeđen iz razreda *Controller*)

1. Preglednik nudi odabir mape za snimanje datoteke
2. Sugerirati pregledniku da isporučeni sadržaj prikaže unutar preglednika

➤ content-disposition:inline

➤ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Drzave() {  
    ...  
    byte[] pdf = report.GenerateAsByteArray();  
    if (pdf != null) {  
        Response.Headers.Add("content-disposition",  
            "inline; filename=drzave.pdf");  
        return File(pdf, "application/pdf");  
        //return File(pdf, "application/pdf", "drzave.pdf");  
        // samo gornji redak otvara Save as dialog  
    }  
    else return NotFound();  
}
```

# Stupci s posebnim predloškom

10


➤ Za svaki artikl prikazana slika u izvješću

➤ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Artikli()  
...  
report.MainTableColumns(columns => {  
    columns.AddColumn(column => {  
        column.PropertyName(nameof(Artikl.SlikaArtikla));  
        ...  
        column.ColumnItemsTemplate(t =>  
            t.ByteArrayImage(string.Empty,  
                fitImages: true));  
    });  
...  
}
```

# Stupci sa sumom vrijednosti u podnožju

11

- Postavlja se konačna suma svih artikala i tekst ispred sume
  - moguće postaviti i međusume za svaku stranicu *PageSummarySettings*
  - Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Artikli()
...
report.MainTableSummarySettings(summarySettings => {
    summarySettings.OverallSummarySettings("Ukupno");
});
report.MainTableColumns(columns => {
    columns.AddColumn(column => {
        column.PropertyName<Artikl>(x => x.CijArtikla);
        column.ColumnItemsTemplate(template => {
            template.TextBlock();
            template.DisplayFormatFormula(obj =>
                string.Format("{0:C2}", obj));
        });
        column.AggregateFunction(aggregateFunction => {
            aggregateFunction.NumericAggregateFunction(
                AggregateFunction.Sum);
            aggregateFunction.DisplayFormatFormula(...
```

## Primjer izvještaja oblika zaglavlje stavke

- Za svaki dokument ispisani osnovni podaci nakon čega slijede sve stavke
  - Nakon ispisa stavki ispisana suma bez PDV-a
- Stupac *Ukupno* je izračunati stupac

10 najvećih kupnji					
Id dokumenta: 2237		Datum dokumenta: 11.02.2014			
Partner: KBC		Iznos dokumenta: 2.178.811,73 kn			
#	Naziv artikla	Količina	Jedinična cijena	Rabat	Ukupno
1	CD/MP3 radio kazetofon, PHILIP MORRIS PH-AZ1038, 1 x kazeta	1,00	289,00 kn	5,00%	274,55 kn
2	DVD/CD/MP3/DivX player, linijski, BIONEER DV-300-K + 7 crtiča, crni	10,00	549,00 kn	9,00%	4.995,90 kn
3	GPS uređaj GURMAN GPS 72	1,00	1.291,00 kn	7,00%	1.200,63 kn
4	Knjiga "Access 2003 za Windows"	2,00	157,00 kn	9,00%	285,74 kn
5	Knjiga "Skok u Linux"	2,00	199,00 kn	2,00%	390,04 kn
6	MBO EVGA, s. 775, 650i Ultra, NVIDIA nForce 650i Ultra, BUS 1333 MHz, serial ATA II, RAID, 7.1 zvuk, 1Gbps, SSSR 2, ATX 2	1,00	684,00 kn	1,00%	677,16 kn
7	Mikrovalna pećnica LGG MS-1924V, 19 litara	1,00	419,00 kn	1,00%	414,81 kn
8	Mobilni uređaj MOTOCOLA V3, GPRS, EMS, Bluetooth, ekran u boji	1,00	999,00 kn	2,00%	979,02 kn
9	Mobilni uređaj NUKIA N73, GPRS, Bluetooth, MMS, FM Radio, kamera, ekran u boji, crveni	2,00	3.499,00 kn	1,00%	6.928,02 kn
10	MP3/WMA player, DESTRUCTIVE Zen Vision M 60 GB, crni	3,00	2.299,00 kn	6,00%	6.483,18 kn
11	MP3/WMA/FM player, CS, MC309F, 1 GB	5,00	403,00 kn	1,00%	1.994,85 kn
12	Notebook ACER Aspire 5610 NWLMi, Kintel Solo Core T1350(1.83ghz), TFT 15.4" WXGA, 1GB (512MB + 512MB POKLON), DVD±RW, HDD 60 GB, bežična mreža, Linux (LX.AU60C.004)	4,00	4.799,00 kn	7,00%	17.852,28 kn
13	Notebook PH Compaq 6715b (GB833EA), KPD Sempron 3800+ (2.2GHz), 15.4" WXGA (1280x800), RAM 1GB SSSR2, HDD 120GB SATA, DVD±RW DL, ATI Radeon X1250, modem, LAN, WLAN, BT, Windows Vista Basic	4,00	5.192,00 kn	8,00%	19.106,56 kn
14	PC računalo HGspotVR PREMIUM II + Windows VISTA Home Premium, KPD Athlon 64 X2 4400+, 1 GB SSSR2, HDD 250 GB SATA-II, GeForce 8500 GT 256 MB, DVD±RW, TV Tuner+zvučnici 2.1 + 19" LCD BENQ monitor + poklon Chat Pack i BITDEFENDER Internet Security 2008	2,00	5.199,00 kn	10,00%	9.358,20 kn
15	RAM, 1 GB, SSSR 2, PC-5300, 667 MHz,	10,00	178,00 kn	2,00%	1.744,40 kn
16	Torba za notebook Marasst MFL30	4,00	141,00 kn	8,00%	518,88 kn
17	Torbica za Strukkle triPOD trio nano 3-1, kožna, siva	3,00	114,00 kn	4,00%	328,32 kn
18	TV LCD 82 cm, SHARP LC32WD1E, 16:9, stereo, HD Ready, 2 x HDMI, DVB -T	1,00	7.399,00 kn	9,00%	6.733,09 kn
19	TV LCD 94 cm, PHILIP MORRIS 37PFL9732, FULL HD, DVB-T	100,00	16.999,00 kn	1,00%	1.682.901,00 kn
Ukupno					763.166,63 kn

# Oblik podataka za izvješća s grupiranjem

13

➤ Potrebno koristiti denormalizirane podatke

➤ Primjer:  Web \ Firma.Mvc \ ViewModels \ StavkaDenorm.cs

```
public class StavkaDenorm {  
    public string OIB { get; set; }  
    public string NazPartnera { get; set; }  
    public int IdDokumenta { get; set; }  
    public DateTime DatDokumenta { get; set; }  
    public decimal IznosDokumenta { get; set; }  
    public int IdStavke { get; set; }  
    public int SifArtikla { get; set; }  
    public decimal KolArtikla { get; set; }  
    public decimal JedCijArtikla { get; set; }  
    public decimal PostoRabat { get; set; }  
    public string NazArtikla { get; set; }  
    [NotMapped] public string UrlDokumenta { get; set; }  
}
```

➤ Svojstvo *UrlDokumenta* predstavljat će adresu dokumenta

➤ Formirat će se nakon dohvata na osnovu usmjeravanja i *IdDokumenta*

➤ Ne postoji u bazi podataka pa se mora označiti s *NotMapped*

# Funkcija za dohvat n najboljih kupnji

14


## ➤ Ispis n dokumenata s najvećim iznosom

- Složen upit te je izveden korištenjem funkcije na BP
- U kontekst potrebno dodati DbSet<StavkaDenorm> i navesti podatak koji jednoznačno određuje podatak

```
CREATE FUNCTION [dbo].[fn_NajveceKupnje] (@N int)
RETURNS TABLE AS
RETURN(
    SELECT P.OIB, P.Naziv AS NazPartnera,
           D.IdDokumenta, D.DatDokumenta, D.IznosDokumenta,
           S.IdStavke, S.SifArtikla, S.KolArtikla,
           S.JedCijArtikla, S.PostoRabat, A.NazArtikla
    FROM
        (SELECT TOP (@N) * FROM Dokument ORDER BY IznosDokumenta DESC) D
    INNER JOIN vw_Partner P ON P.IdPartnera = D.IdPartnera
    LEFT OUTER JOIN Stavka S ON S.IdDokumenta = D.IdDokumenta
    LEFT OUTER JOIN Articl A ON A.SifArtikla = S.SifArtikla
)
```

# Poziv funkcije s parametrima


15

- Za dohvat iz DbSet-a koji se odnosi na funkciju, potrebno napisati SQL upit (u ovom slučaju parametrizirani)
- Podaci moraju biti poredani po stupcima po kojima se želi izvršiti grupiranje
  - Poredak nakon toga po želji
- Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Dokumenti() {  
    int n = 10;  
    var param = new SqlParameter("N", n);  
    string naslov = $"{n} najvećih kupnji";  
    var stavke = await ctx.StavkaDenorm  
        .AsNoTracking()  
        .FromSqlRaw(  
            "SELECT * FROM fn_NajveceKupnje(@N)", param)  
        .OrderBy(s => s.IdDokumenta)  
        .ThenBy(s => s.NazArtikla)  
        .ToListAsync();  
}
```

# Postavljanje adrese za ažuriranje dokumenta

16


- Unutar PDF-a bit će dodana poveznica za ažuriranje dokumenta
- Potrebno pripremiti svojstvo s adresom
  - Akcija Edit na upravljaču Dokument s parametrom id
  - Sprema se u svojstvo UrlDokumenta za svaki element dohvaćene liste
- Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Dokumenti() {  
    ...  
    var stavke = await ctx.StavkaDenorm  
        ...  
        .ToListAsync();  
  
    stavke.ForEach(s => s.UrlDokumenta = Url.Action("Edit",  
                                                    "Dokument",  
                                                    new { id = s.IdDokumenta }  
    ));  
}
```



# Definiranje stupaca po kojima se vrši grupiranje

17

- Prilikom definiranja stupaca u prikazu, prvo se navode stupci po kojima se vrši grupiranje
  - Neće biti prikazani u tablici, već iznad nje
- Potrebno napisati kriterij pripadaju li dvije vrijednosti tog „stupca” istoj grupi
- Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Artikli()  
...  
report.MainTableColumns(columns => {  
    columns.AddColumn(column => {  
        column.PropertyName("IdDokumenta");  
        column.Group(  
            (val1, val2) => {  
                return (int)val1 == (int)val2;  
            });  
    });  
});
```

# Stupac s automatskim izračunom vrijednosti

18


➡ Vrijednost stupca s ukupnom cijenom neke stavke računa se na osnovu ostalih vrijednosti iz trenutnog retka

➡ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Artikli()  
...  
report.MainTableColumns(columns => {  
    columns.AddColumn(column => {  
        column.CalculatedField(  
            list => {  
                if (list == null) return string.Empty;  
                decimal kolArtikla = (decimal)list.GetValueOf("KolArtikla");  
                decimal postoRabat = (decimal)list.GetValueOf("PostoRabat");  
                decimal jedCijArtikla = (decimal)list.GetValueOf("JedCijArtikla");  
                var iznos = jedCijArtikla * kolArtikla * (1 - postoRabat);  
                return iznos;  
            });  
        });  
    });  
});
```

# Zaglavlje grupe

19

- Kao predložak za zaglavlje pojedine grupe potrebno napisati vlastiti predložak kojim se definira kako izgleda zaglavlje izvještaja i zaglavlje pojedine grupe
- Predložak se postavlja postupkom *CustomHeader* u *PagesHeader*
  - Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public async Task<IActionResult> Artikli()  
...  
report.PagesHeader(header => {  
    header.CacheHeader(cache: true);  
    header.CustomHeader(new MasterDetailsHeaders(naslov)  
    {  
        PdfRptFont = header.PdfFont  
    });  
});
```

# Predložak za zaglavlje izvještaja

20

➡ Zaglavlje izvještaja sastoji se od jednog retka s naslovom

➡ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public class MasterDetailsHeaders : IPageHeader {  
    private string naslov;  
    public MasterDetailsHeaders(string naslov) {  
        this.naslov = naslov;  
    }  
    public IPdfFont PdfRptFont { set; get; }  
  
    public PdfGrid RenderingReportHeader(...  
    {  
        var table = new PdfGrid(numColumns: 1) { WidthPercentage = 100 };  
        table.AddSimpleRow(  
            (cellData, cellProperties) => {  
                cellData.Value = naslov;  
                ...  
            });  
        return table.AddBorderToTable();  
    }  
}
```

# Predložak za zaglavlje grupe

21

➤ Vrijednosti grupe mogu se dobiti iz ulaznog argumenta tipa *ICollection<CellData>*

➤ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public class MasterDetailsHeaders : IPager {
    public PdfGrid RenderingGroupHeader(...
        ICollection<CellData> newGroupInfo ...){

        var idDokumenta = newGroupInfo
            .GetSafeStringValueOf("IdDokumenta");

        ...
        var iznosDokumenta = (decimal) newGroupInfo
            .GetValueOf("IznosDokumenta");
    }
}
```

# Predložak za zaglavlje grupe

22

➤ Zaglavlje grupe definirano kao tablica s 4 stupca s podacima grupe

➤ Svaka ćelija može sadržavati fiksni tekst ili biti vezana za neku vrijednost

➤ Primjer:  Web \ Firma.Mvc \ Controllers \ ReportController.cs

```
public class MasterDetailsHeaders : IPageHeader {
    public PdfGrid RenderingGroupHeader( ...
        ...
        var table = new PdfGrid(relativeWidths: new[]
            { 2f, 5f, 2f, 3f }) { WidthPercentage = 100 };

        table.AddSimpleRow(
            ...
            (cellData, cellProperties) => {
                cellData.Value = datDokumenta;
                cellProperties.PdfFont = PdfRptFont;
                cellProperties.HorizontalAlignment =
                    HorizontalAlignment.Left;

                cellProperties.DisplayFormatFormula =
                obj => ((DateTime)obj).ToString("dd.MM.yyyy");
            });
}
```