Fast-Report

Open-Source

หัวข้อบรรยาย

1. Fast-Report คืออะไร
2. Implementations
3. Designer
4. Workshop
5. Reference

Fast-Report คืออะไร

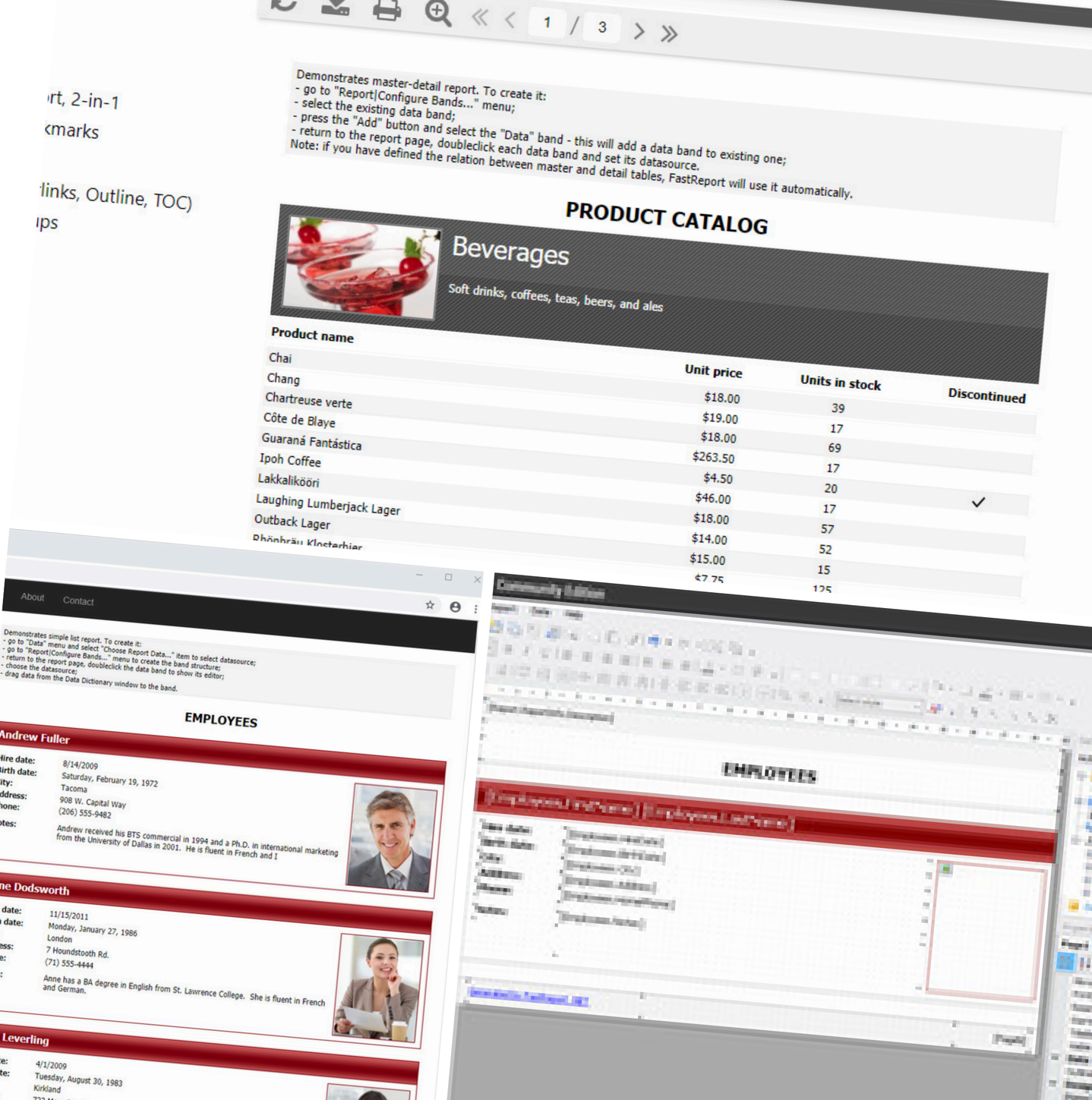
Tools ที่จะมาช่วยเราในการออกรายงานได้ ฟรี ง่าย และรวดเร็วขึ้น

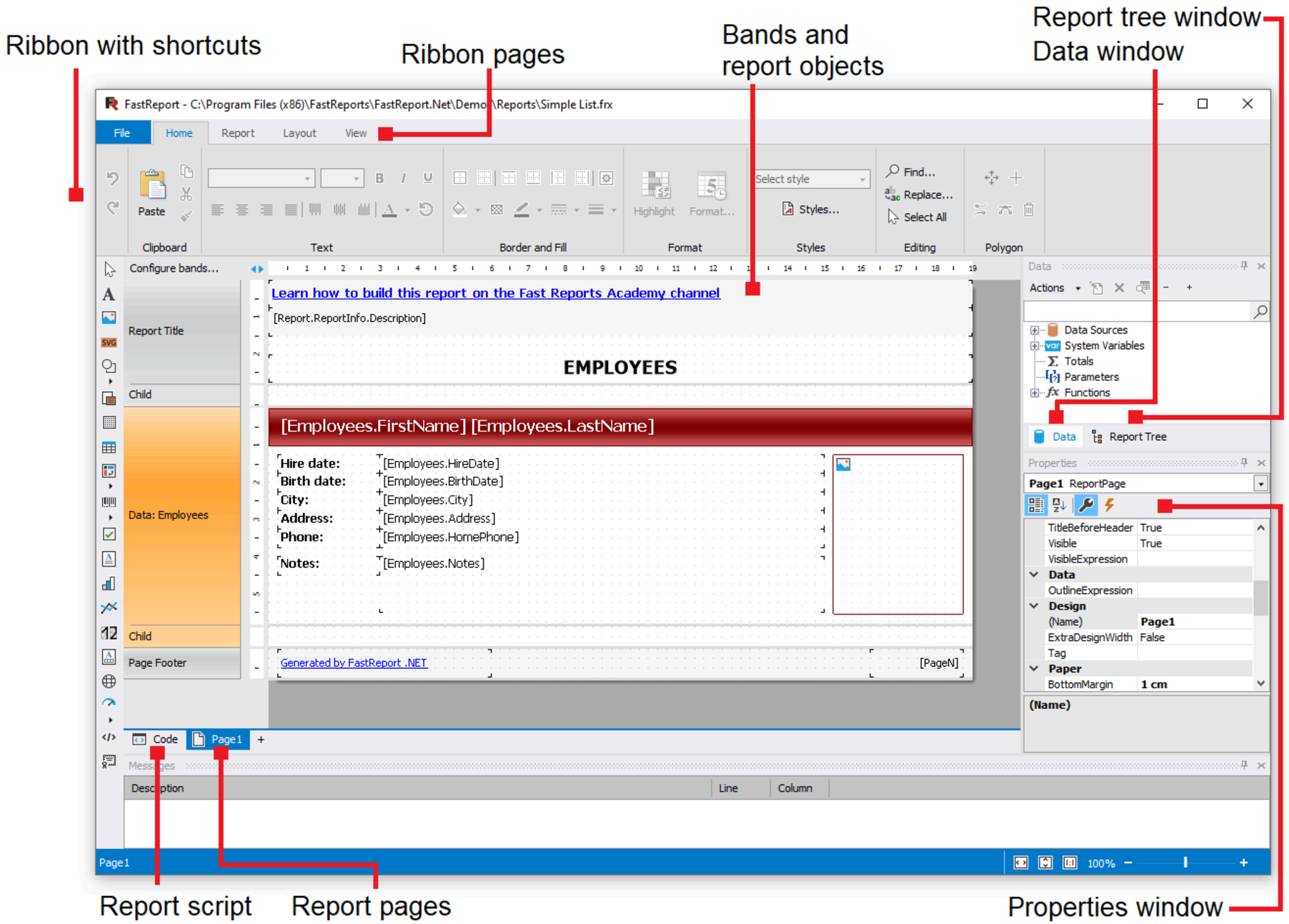
Open source report generator สำหรับ

- .NET 6
- .NET Core
- .NET Framework.

สามารถใช้ได้กับ

- MVC
- Web API
- Console Applications

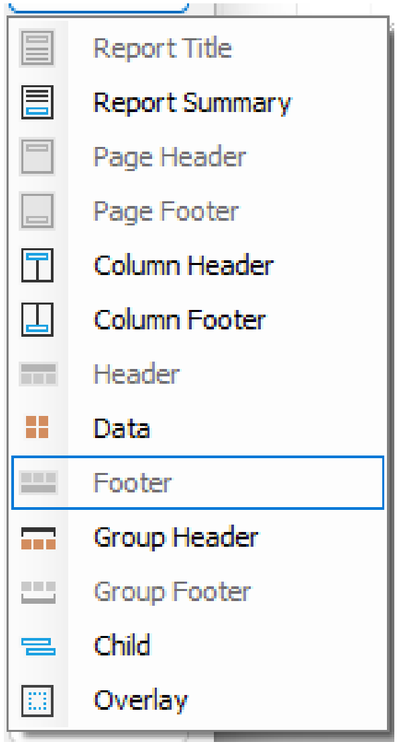




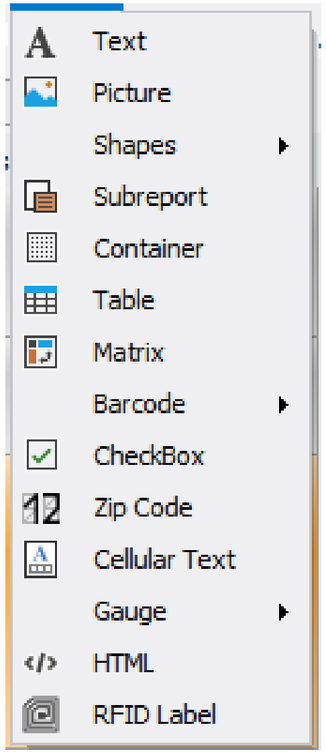
Report Designer

Open Source Designer

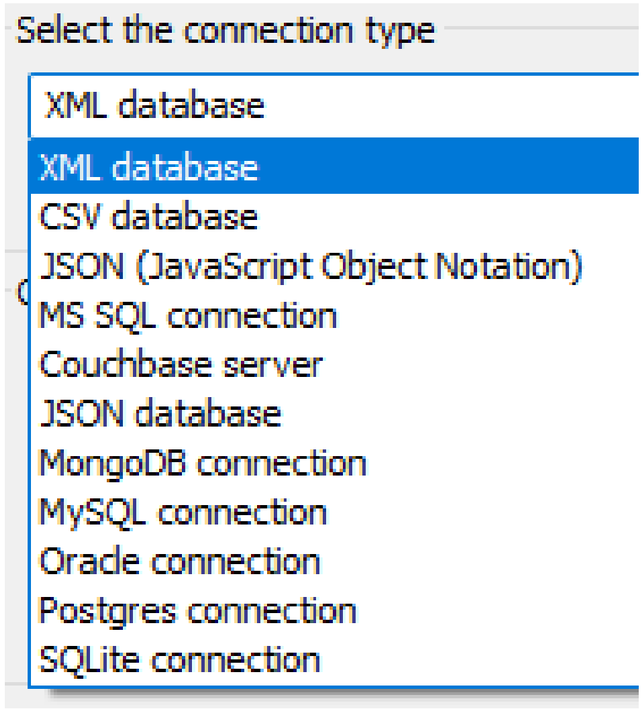
Band-oriented: 13 types



Objects: 8 types



DataSource: 11 types



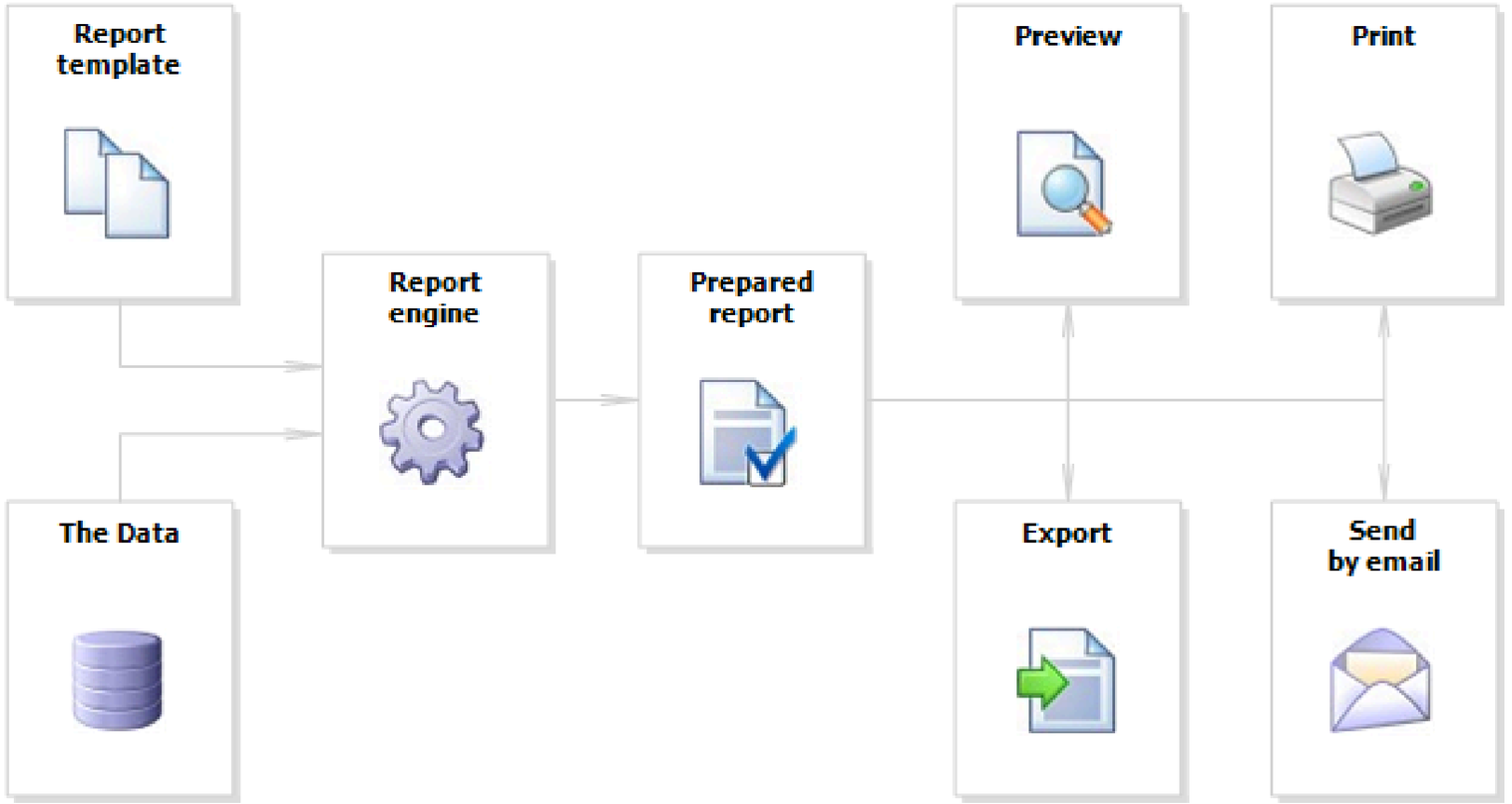
+ BusinessObjectDataSource

Export: 8 types

- HTML
- BMP
- PNG
- JPEG
- GIF
- TIFF
- EMF
- PDF

- FastReport has ability to get data from business objects of IEnumerable type.

Implementations



ReportTemplateI.ftx

```
<?xml version="1.0" encoding="utf-8"?>
<Report ScriptLanguage="CSharp" ReportInfo.Name="testName" ReportInfo.Author="testAuthor"
ReportInfo.Version="1.0.1" ReportInfo.Description="testDescription" ReportInfo.Created="04/02/2025
14:10:09" ReportInfo.Modified="04/04/2025 16:17:57" ReportInfo.CreatorVersion="2024.2.0.0">
  <Dictionary>
    <BusinessObjectDataSource Name="JSON"
DataType="System.Collections.Generic.List`1[SimpleFastReport.API.DTOs.EmployeeReponseDTO,
SimpleFastReport.API, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null]" Enabled="true">
      <Column Name="EmployeeId" DataType="System.Double"/>
      <Column Name="FirstName" DataType="System.String"/>
      <Column Name="LastName" DataType="System.String"/>
    </BusinessObjectDataSource>
  </Dictionary>
  <ReportPage Name="Page2" Watermark.Font="Arial, 60pt">
    <OverlayBand Name="Overlay1" Width="718.2" Height="935.55" Guides="387.45,387.45">
      <TextObject Name="Text12" Width="718.2" Height="85.05" CanGrow="true" GrowToBottom="true"
CanBreak="false" Text="[UpperCase(&quot;Employee list&quot;)]" Padding="2, 2, 2, 2" AutoWidth="true"
HorzAlign="Center" VertAlign="Center" Font="Leelawadee, 20pt"/>
    </OverlayBand>
  </ReportPage>
  <ReportPage Name="Page1" Watermark.Font="Arial, 60pt" ResetPageNumber="true">
    <ReportTitleBand Name="ReportTitle1" Width="718.2" Height="37.8"
Guides="236.25,236.25,236.25,236.25,236.25">
      <TextObject Name="Text7" Width="718.2" Height="37.8" Text="[UpperCase(&quot;Employees&quot;)]"
HorzAlign="Center" VertAlign="Center" Font="Leelawadee, 18pt, style=Bold"/>
      <ChildBand Name="Child1" Top="41.8" Width="718.2" Height="18.9"/>
    </ReportTitleBand>
    <GroupHeaderBand Name="GroupHeader1" Top="64.7" Width="718.2" Height="37.8" Fill="LinearGradient"
Fill.StartColor="LightCoral" Fill.EndColor="Maroon" Fill.Angle="270" Fill.Focus="0.5" Fill.Contrast="1"
Condition="[JSON.EmployeeId]" KeepTogether="true">
      <TextObject Name="Text11" Left="18.9" Width="680.4" Height="37.8" Text="[JSON.FirstName]
[JSON.LastName]" VertAlign="Center" Font="Leelawadee, 16pt, style=Bold" TextFill.Color="White"/>
    </GroupHeaderBand>
    <PageFooterBand Name="PageFooter1" Top="276.15" Width="718.2" Height="18.9">
      <TextObject Name="Text1" Left="614.25" Width="94.5" Height="18.9" Text="[PageN] / [TotalPages#]"
Font="Arial, 10pt"/>
    </PageFooterBand>
  </ReportPage>
</Report>
```

+

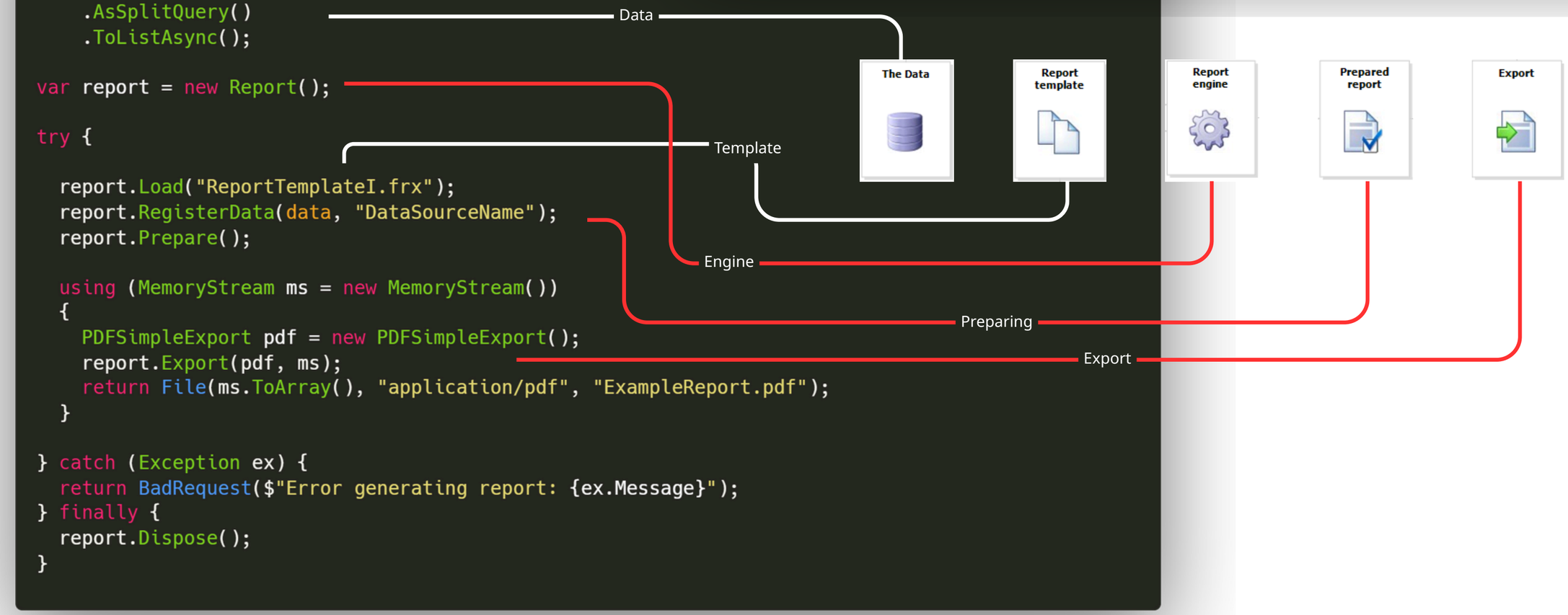
ReportDataI.json


```
[
  {
    "employeeId": 1,
    "firstName": "John",
    "lastName": "Doe",
    "email": "john.doe@example.com",
    "phone": "123-456-7890",
    "hireDate": "2022-06-15T00:00:00",
    "salary": 55000,
    "department": "IT",
    "imageUrl": "https://lineagentapi.uatsiamsmile.com/messageImages/030425/image376369_20250403_095648.jpg"
  },
  ...
]
```

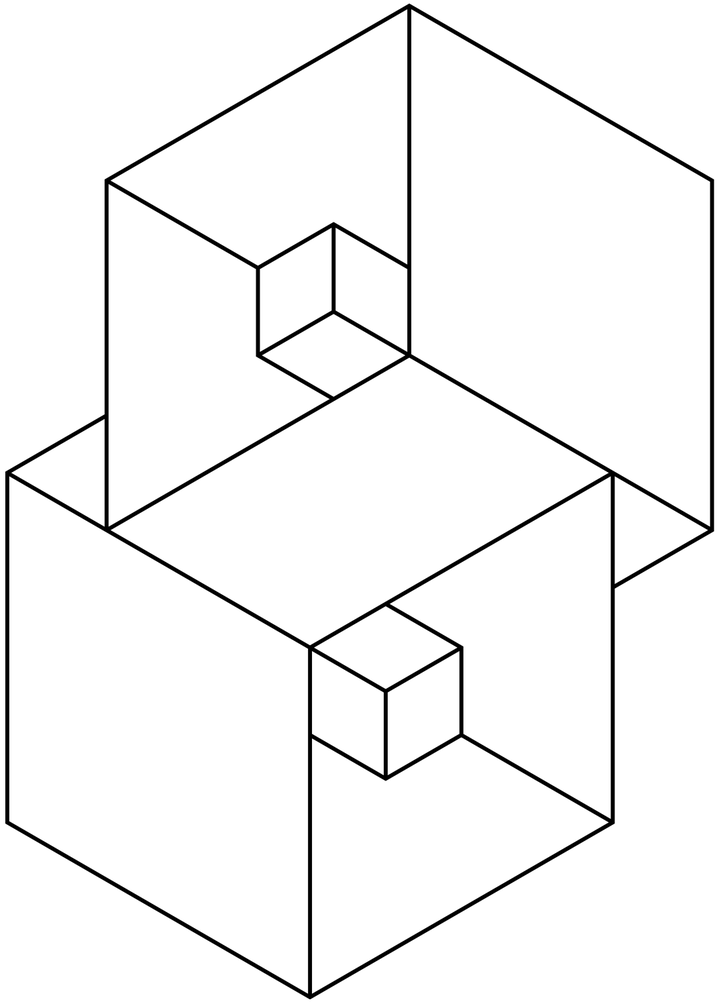
Report Engine

```
<PackageReference Include="FastReport.OpenSource" Version="2025.1.0" />
<PackageReference Include="FastReport.OpenSource.Export.PdfSimple" Version="2025.1.0" />
<PackageReference Include="FastReport.OpenSource.Data.Json" Version="2021.4.0" />
<PackageReference Include="FastReport.OpenSource.Web" Version="2025.1.0" />
```

```
var data = await _dBContext.Products
    .Include(_ => _.ProductGroup)
    .Where(_ => _.IsActive == true)
    .AsSplitQuery()
    .ToListAsync();

var report = new Report();

try {

  report.Load("ReportTemplateI.frx");
  report.RegisterData(data, "DataSourceName");
  report.Prepare();

  using (MemoryStream ms = new MemoryStream())
  {
    PDFSimpleExport pdf = new PDFSimpleExport();
    report.Export(pdf, ms);
    return File(ms.ToArray(), "application/pdf", "ExampleReport.pdf");
  }

} catch (Exception ex) {
  return BadRequest($"Error generating report: {ex.Message}");
} finally {
  report.Dispose();
}
```

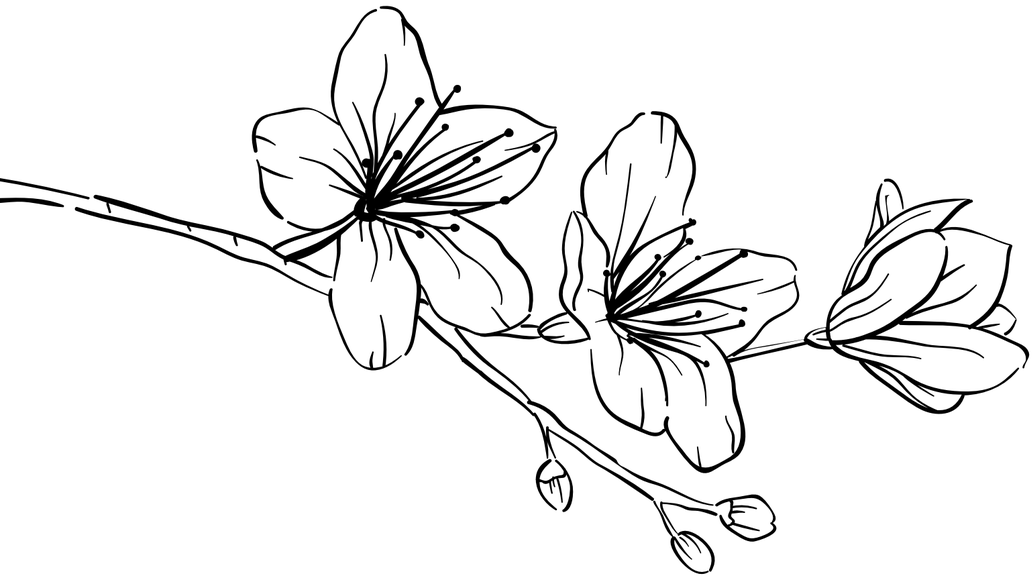


Workshop



Reference

Source Code: https://github.com/Pratchaya0/csharp-fast-report.git
Fast-Report Manual: https://www.fast-report.com/public_download/docs/FRNet/online/en/index.html
Fast-Report Repository: https://github.com/FastReports/FastReport
Fast-Report OpenSource v2025.2.0: https://github.com/FastReports/FastReport/releases/tag/v2025.2.0



ขอบคุณครับ