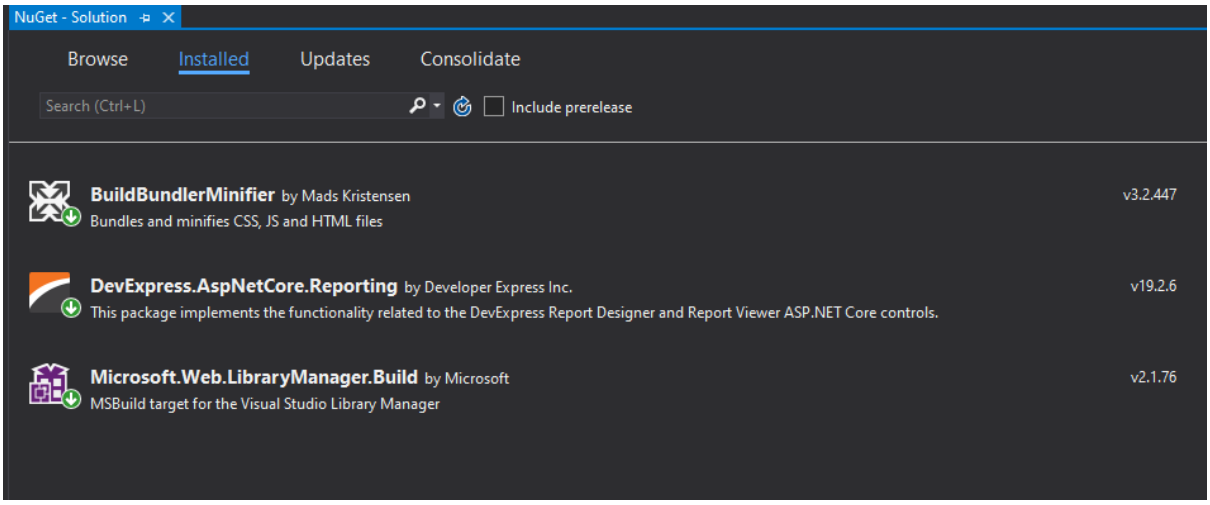
How add DevExpress Reporting to ASP.NET Core 3 Project

1. Install below package to project



BuildBundlerMinifier by Mads Kristensen v3.2.447
Bundles and minifies CSS, JS and HTML files

DevExpress.AspNetCore.Reporting by Developer Express Inc. v19.2.6
This package implements the functionality related to the DevExpress Report Designer and Report Viewer ASP.NET Core controls.

Microsoft.Web.LibraryManager.Build by Microsoft v2.1.76
MSBuild target for the Visual Studio Library Manager

2. Add 'package.json' on project, with below data

```
{
  "name": "asp.net",
  "version": "1.0.0",
  "license": "MIT",
  "private": true,
  "dependencies": {
    "bootstrap": "^4.3.1",
    "cldrjs": "^0.5.0",
    "devextreme": "19.2-next",
    "devexpress-richedit": "19.2-next",
    "@devexpress/analytics-core": "19.2-next",
    "devexpress-reporting": "19.2-next",
    "globalize": "^1.3.0",
    "jquery-ui-dist": "^1.12.1"
  }
}
```

3. Add ‘bundleconfig.json’ on project, with below data

```
[
  {
    "outputFileName": "wwwroot/css/thirdparty.bundle.css",
    "inputFiles": [
      "node_modules/jquery-ui-dist/jquery-ui.min.css",
      "node_modules/bootstrap/dist/css/bootstrap.min.css",
      "node_modules/devextreme/dist/css/dx.common.css",
      "node_modules/devextreme/dist/css/dx.light.css"
    ],
    "minify": {
      "enabled": false,
      "adjustRelativePaths": false
    }
  },
  {
    "outputFileName": "wwwroot/css/viewer.part.bundle.css",
    "inputFiles": [
      "node_modules/@devexpress/analytics-core/dist/css/dx-analytics.common.css",
      "node_modules/@devexpress/analytics-core/dist/css/dx-analytics.light.css",
      "node_modules/devexpress-reporting/dist/css/dx-webdocumentviewer.css"
    ],
    "minify": {
      "enabled": false,
      "adjustRelativePaths": false
    }
  },
  {
    "outputFileName": "wwwroot/css/designer.part.bundle.css",
    "inputFiles": [
      "node_modules/devexpress-richedit/dist/dx.richedit.css",
      "node_modules/@devexpress/analytics-core/dist/css/dx-querybuilder.css",
      "node_modules/devexpress-reporting/dist/css/dx-reportdesigner.css"
    ],
    "minify": {
      "enabled": false,
      "adjustRelativePaths": false
    }
  },

  {
    "outputFileName": "wwwroot/js/thirdparty.bundle.js",
    "inputFiles": [
      "node_modules/jquery/dist/jquery.min.js",
      "node_modules/jquery-ui-dist/jquery-ui.min.js",
      "node_modules/knockout/build/output/knockout-latest.js",
      "node_modules/bootstrap/dist/js/bootstrap.min.js",
      "node_modules/cldrjs/dist/cldr.js",
      "node_modules/cldrjs/dist/cldr/event.js",
      "node_modules/cldrjs/dist/cldr/supplemental.js",
      "node_modules/cldrjs/dist/cldr/unresolved.js",
      "node_modules/globalize/dist/globalize.js",
      "node_modules/globalize/dist/globalize/message.js",
      "node_modules/globalize/dist/globalize/number.js",
      "node_modules/globalize/dist/globalize/currency.js",
      "node_modules/globalize/dist/globalize/date.js",
      "node_modules/devextreme/dist/js/dx.all.js",

      "node_modules/ace-builds/src-min-noconflict/ace.js",
      "node_modules/ace-builds/src-min-noconflict/ext-language_tools.js",
      "node_modules/ace-builds/src-min-noconflict/theme-dreamweaver.js",
      "node_modules/ace-builds/src-min-noconflict/theme-ambiance.js",
      "node_modules/ace-builds/src-min-noconflict/snippets/text.js"
    ],
    "minify": {
      "enabled": false
    },
    "sourceMap": false
  },
  {
    "outputFileName": "wwwroot/js/viewer.part.bundle.js",
    "inputFiles": [
      "node_modules/@devexpress/analytics-core/dist/js/dx-analytics-core.min.js",
      "node_modules/devexpress-reporting/dist/js/dx-webdocumentviewer.min.js"
    ],
    "minify": {
      "enabled": false
    },
    "sourceMap": false
  },


  {
    "outputFileName": "wwwroot/js/designer.part.bundle.js",
    "inputFiles": [
      "node_modules/jszip/dist/jszip.min.js",
      "node_modules/devexpress-richedit/dist/dx.richedit.min.js",

      "node_modules/@devexpress/analytics-core/dist/js/dx-querybuilder.min.js",
      "node_modules/devexpress-reporting/dist/js/dx-reportdesigner.min.js"
    ],
    "minify": {
      "enabled": false
    },
    "sourceMap": false
  }
]
```

4. Add ‘libman.json’ on project, with below data

```
{
  "version": "1.0",
  "defaultProvider": "cdnjs",
  "libraries": [
    {
      "provider": "filesystem",
      "library": "node_modules/devextreme/dist/css/icons/",
      "destination": "wwwroot/css/icons",
      "files": [
        "dxicons.ttf",
        "dxicons.woff2",
        "dxicons.woff"
      ]
    }
  ]
}
```

5. Add folder ‘Services’ on project
6. Add class ‘ReportStorageWebExtension.cs’ on ‘Services’ folder, with add in below code on class

```
public class ReportStorageWebExtension1 : DevExpress.XtraReports.Web.Extensions.ReportStorageWebExtension
{
    readonly string ReportDirectory;
    const string FileExtension = ".repx";
    public ReportStorageWebExtension1(IWebHostEnvironment env)
    {
        ReportDirectory = Path.Combine(env.ContentRootPath, "Reports");
        if (!Directory.Exists(ReportDirectory))
        {
            Directory.CreateDirectory(ReportDirectory);
        }
    }

    public override bool CanSetData(string url)
    {
        // Determines whether or not it is possible to store a report by a given URL.
        // For instance, make the CanSetData method return false for reports that should be read-only in your storage.
        // This method is called only for valid URLs (i.e., if the IsValidUrl method returned true) before the SetData method is called.

        return true;
    }

    public override bool IsValidUrl(string url)
    {
        // Determines whether or not the URL passed to the current Report Storage is valid.
        // For instance, implement your own logic to prohibit URLs that contain white spaces or some other special characters.
        // This method is called before the CanSetData and GetData methods.

        return true;
    }

    public override byte[] GetData(string url)
    {
        // Returns report layout data stored in a Report Storage using the specified URL.
        // This method is called only for valid URLs after the IsValidUrl method is called.
        try
        {
            if (Directory.EnumerateFiles(ReportDirectory).Select(Path.GetFileNameWithoutExtension).Contains(url))
            {
                return File.ReadAllBytes(Path.Combine(ReportDirectory, url + FileExtension));
            }
            throw new DevExpress.XtraReports.Web.ClientControls.FaultException(string.Format("Could not find report '{0}'.", url));
        }
        catch (Exception)
        {
            throw new DevExpress.XtraReports.Web.ClientControls.FaultException(string.Format("Could not find report '{0}'.", url));
        }
    }

    public override Dictionary<string, string> GetUrls()
    {
        // Returns a dictionary of the existing report URLs and display names.
        // This method is called when running the Report Designer,
        // before the Open Report and Save Report dialogs are shown and after a new report is saved to a storage.

        return Directory.GetFiles(ReportDirectory, "*" + FileExtension)
                                 .Select(Path.GetFileNameWithoutExtension)
                                 .ToDictionary<string, string>(x => x);
    }

    public override void SetData(XtraReport report, string url)
    {
        // Stores the specified report to a Report Storage using the specified URL.
        // This method is called only after the IsValidUrl and CanSetData methods are called.
        report.SaveLayoutToXml(Path.Combine(ReportDirectory, url + FileExtension));
    }

    public override string SetNewData(XtraReport report, string defaultUrl)
    {
        // Stores the specified report using a new URL.
        // The IsValidUrl and CanSetData methods are never called before this method.
        // You can validate and correct the specified URL directly in the SetNewData method implementation
        // and return the resulting URL used to save a report in your storage.
        SetData(report, defaultUrl);
        return defaultUrl;
    }
}
```

7. Modify ‘Startup.cs’ ‘ConfigureServices’ function like below

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDevExpressControls();
    services.AddScoped<ReportStorageWebExtension, ReportStorageWebExtension1>();

    services.AddControllersWithViews();

    services.ConfigureReportingServices(configurator => {
        configurator.ConfigureReportDesigner(designerConfigurator => {
            designerConfigurator.RegisterDataSourceWizardConfigFileConnectionStringsProvider();
        });
        configurator.ConfigureWebDocumentViewer(viewerConfigurator => {
            viewerConfigurator.UseCachedReportSourceBuilder();
        });
    });
}
```

8. Modify ‘Startup.cs’ ‘Configure’ function like below

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    DevExpress.XtraReports.Configuration.Settings.Default.UserDesignerOptions.DataBindingMode = DevExpress.XtraReports.UI.DataBindingMode.Expressions;
    app.UseDevExpressControls();
    System.Net.ServicePointManager.SecurityProtocol |= System.Net.SecurityProtocolType.Tls12;
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
        app.UseHsts();
    }
    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthorization();
    app.UseEndpoints(endpoints => {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
}
```

9. Modify ‘Views\Shared_Layout.cshtml’ header like below

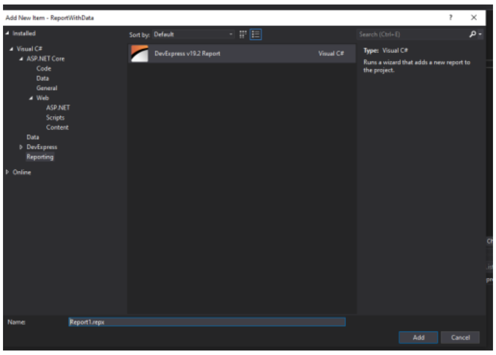
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - ReportWithData</title>
    <link rel="stylesheet" href="~/css/thirdparty.bundle.css" />
    <link rel="stylesheet" href="~/css/site.css" />
    <script src="~/js/thirdparty.bundle.js"></script>
    <script src="~/js/site.js" asp-append-version="true"></script>
</head>
```

10. Modify ‘Views\Shared_Layout.cshtml’ footer like below

```
    <footer class="border-top footer text-muted">
        <div class="container">
            &copy; 2020 - ReportWithData - <a asp-area="" asp-controller="Home" asp-
action="Privacy">Privacy</a>
        </div>
    </footer>
    @RenderSection("Scripts", required: false)
</body>
</html>
```

Sample for add Viewer and Designer

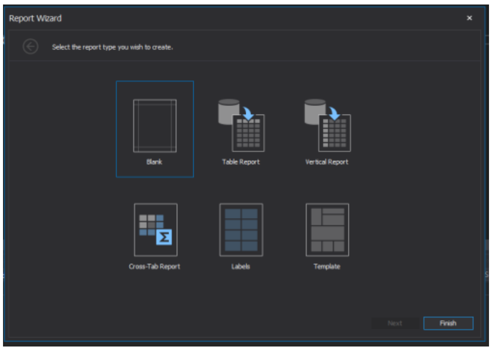
1. Add folder ‘Reports on project (to store all report file)
2. Add Blank XtraReport on folder ‘Reports’



Type: Visual C#

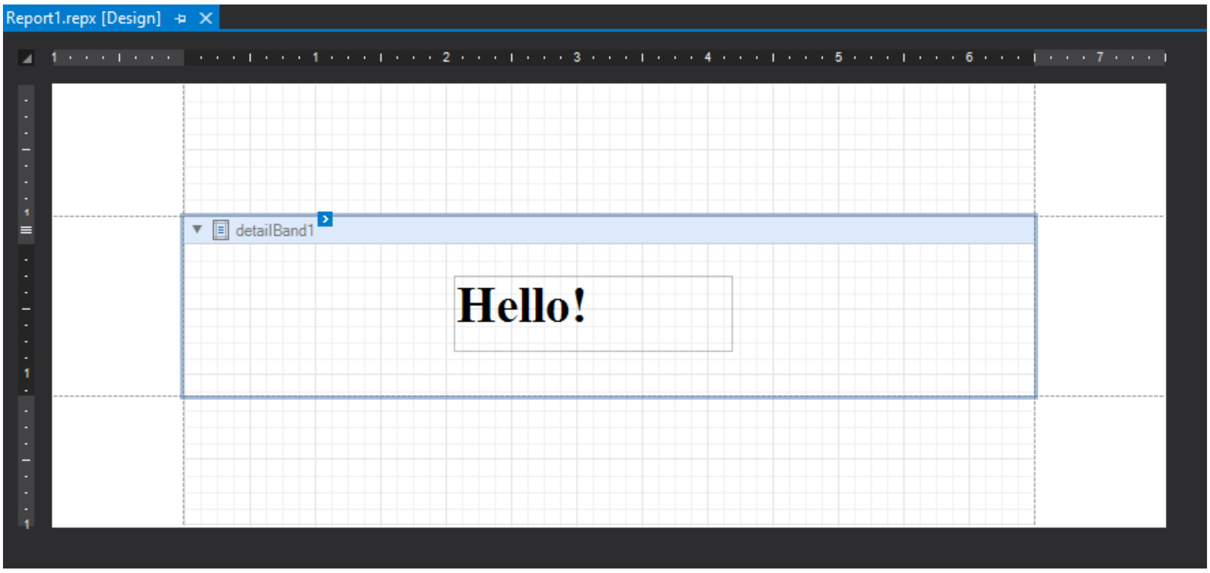
Runs a wizard that adds a new report to the project.

Name: Report1.repx



Select the report type you wish to create.

3. Add hello label on Detail Band



4. Add below two method on ‘HomeController’ (optional, add sample call)

```
public IActionResult Viewer()
{
    return View();
}

public IActionResult Designer()
{
    return View();
}
```

5. Add view page ‘Viewer.cshtml’ on ‘Views\Home’, with below code

```
@using DevExpress.AspNetCore

<link rel="stylesheet" href="~/css/viewer.part.bundle.css" />

<script src="~/js/viewer.part.bundle.js"></script>

@Html.DevExpress().WebDocumentViewer("DocumentViewer").Height("1000px").Bind("Report1")
```

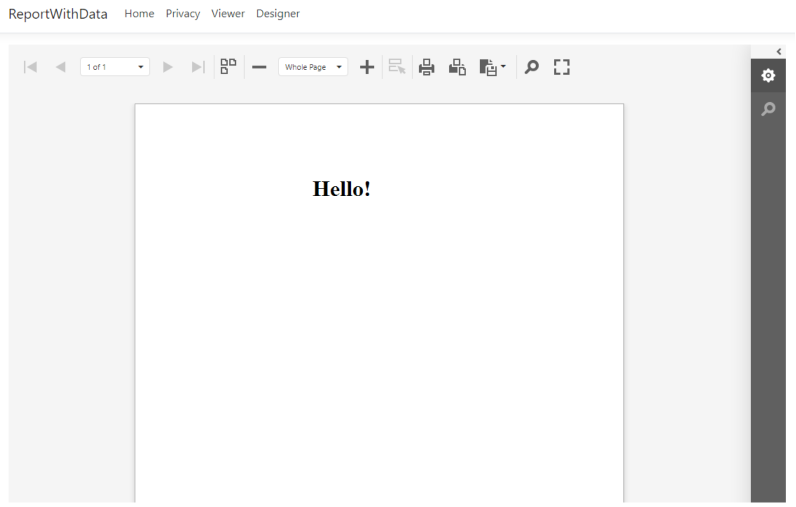
6. Add view page ‘Designer.cshtml’ on ‘Views\Home’, with below code

```
@using DevExpress.AspNetCore
<link rel="stylesheet" href="~/css/viewer.part.bundle.css" />
<link rel="stylesheet" href="~/css/designer.part.bundle.css" />

<script src="~/js/viewer.part.bundle.js"></script>
<script src="~/js/designer.part.bundle.js"></script>

@(Html.DevExpress().ReportDesigner("reportDesigner")
    .Height("1000px")
    .Bind("Report1"))
```

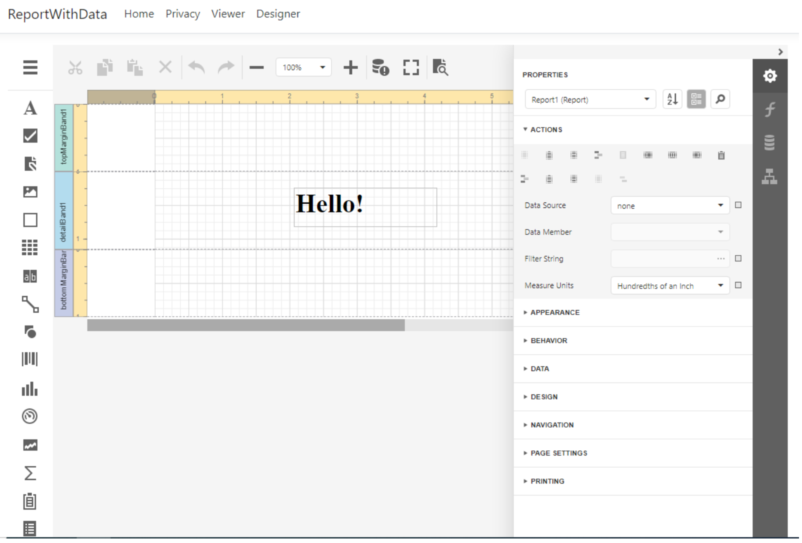
7. Add link on ‘Views\Shared_Layout.cshtml’

```
<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Viewer">Viewer</a>
<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Designer">Designer</a>
```

8. Test Report Viewer



9. Test Report Designer



Report1 (Report)

Data Source: none

Data Member:

Filter String:

Measure Units: Hundredths of an Inch