



Version: 4.2.2

Data Internal Export

Article 08/22/2023

The dReveal grid component provides methods to export its data to a file or stream in PDF, RawData and XLSX format. These methods are available on the server side.

Server Side Export

The methods provided by the grid component on the server side are detailed below:

Name	Return	Format
ExportTo	ActionResult	PDF
ExportTo	ActionResult	RawData
ExportTo	ActionResult	XLSX

These methods are located in the `Utils` class from the namespace `InfoArch.Web.Mvc.Helpers`.

It is important to mention that exists some properties in the `GridSettings` object located into the `GridViewModel` class, for customizing the Excel exportation. These properties are detailed below:

Property	Description
<code>GridSettings.CustomFileName</code>	Excel file name
<code>GridSettings.CustomTitle</code>	The title of the report in the Excel file

Controller

The following code demonstrates how to implement the methods in a controller class:

```
1  [HttpPost]
2  public FileStreamResult
   GridReportPdf([ModelBinder(typeof(DRevealDataStructureBinder))]
   DRevealDataStructure gridState)
3  {
4      // Create GridViewModel
5      GridViewModel viewModel = GetGridViewModel();
6      viewModel.SetState(gridboardState);
7      viewModel.LoadInitialConfiguration();
8
9      // Set config in GridSettings
10     viewModel.GridSettings.CustomFileName = "MyPdfGrid";
11     viewModel.GridSettings.CustomTitle = "Title Grid";
12
13     //Export Excel
14     return Utils.ExportTo(gridState.ExportType, gridViewModel, Response);
15 }
16
17 [HttpPost]
18 public FileStreamResult
   GridReportExcel([ModelBinder(typeof(DRevealDataStructureBinder))]
   DRevealDataStructure gridState)
19 {
20     // Create GridViewModel
21     GridViewModel viewModel = GetGridViewModel();
22     viewModel.SetState(gridboardState);
23     viewModel.LoadInitialConfiguration();
24
25     // Set config in GridSettings
26     viewModel.GridSettings.CustomFileName = "MyExcelGrid";
27     viewModel.GridSettings.CustomTitle = "Title Grid";
28
29     // If value is FALSE, this line is optional
30     viewModel.SetXLExportModeOnlyData(false);
31
32     // Export Excel
33     return Utils.ExportTo(gridState.ExportType, gridViewModel, Response);
34 }
35
36 [HttpPost]
37 public FileStreamResult
   GridReportRawData([ModelBinder(typeof(DRevealDataStructureBinder))]
   DRevealDataStructure gridState)
38 {
```

```

39 // Create GridViewModel
40 GridViewModel viewModel = GetGridViewModel();
41 viewModel.SetState(gridboardState);
42 viewModel.LoadInitialConfiguration();
43 viewModel.SetXLEExportModeOnlyData(true);
44
45 // Set config in GridSettings
46 viewModel.GridSettings.CustomFileName = "MyRawDataGrid";
47 viewModel.GridSettings.CustomTitle = "Title Grid";
48
49 // Export Excel
50 return Utils.ExportTo(gridState.ExportType, gridViewModel, Response);
51 }
52
53 private GridViewModel GetGridViewModel()
54 {
55     return new GridViewModel(
56         memoryStream: fileStream,
57         connectionString: "Data Source=MyServer; Initial Catalog=myDatabase;
User ID=myUser; Password=myPassword",
58         gridId: "grid_id",
59         htmlFormId: "main_form",
60         rowsPerPage: 15,
61         enableColumnFinderFeature: true,
62         isGridPagerSimpleMode: true,
63         fullGridControllerActionPath:
HttpContext.Request.Url.AbsolutePath,
64         gridActionName: string.Empty,
65         drillThroughControllerAction: string.Empty,
66         saveGridControllerAction: string.Empty,
67         exportToExcelGridControllerAction: string.Empty
68     );
69 }

```

Client Side Export

The following code demonstrates how to call to the server side methods:

```

<form id="main_form" method="post">
  <div>
    <button
      type="submit"
      formaction="@Url.Action("GridExportPdf")"
      formmethod="post"
    >

```

```

        name="dr_gridExport"
        value="pdf">Download PDF</button>

<button
    type="submit"
    formaction="@Url.Action("GridExportExcel")"
    formmethod="post"
    name="dr_gridExport"
    value="xlsxXL">Download EXCEL</button>

<button
    type="submit"
    formaction="@Url.Action("GridExportRawData")"
    formmethod="post"
    name="dr_gridExport"
    value="xlsxXL">Download RAW DATA</button>
</div>
</form>

```

Export requests require sending an input field with the name `dr_gridExport`. The supported values are `pdf` for downloading files of type **PDF** and `xlsxXL` for downloading files of type **EXCEL** and **RAW DATA**.

The decision of whether to download **EXCEL** or **RAW DATA** is made by the **GridViewModel** object, using the `SetXLExportModeOnlyData(bool enable)` method. The value `TRUE` indicates that RAW DATA will be downloaded and the value `FALSE` indicates that EXCEL will be downloaded.