Version: 4.2.2

Column Selection Component

Article 08/28/2023

The dReveal column selection component provides features to select the columns displayed in the grid component.

Controller

This model binder structure is mapped via the `DRevealDataStructure` class, that returns the current state of the column selection component. This state contains the selected columns applied. The state received after mapping processing must be assigned to the Grid View Model method called `SetState`.

The following code demonstrates how to invoke the method mentioned before:

```
[Route("GridReport")]
public ActionResult GridReport(
    [ModelBinder(typeof(DRevealDataStructureBinder))] DRevealDataStructure gridState
)
{
    GridViewModel viewModel = new GridViewModel(
        memoryStream: fileStream,
        connectionString: "Data Source=MyServer; Initial Catalog=myDatabase; User ID=myUser; Password=myPassword",
        gridId: "grid_id",
        htmlFormId: "main_form",
        rowsPerPage: 15,
        enableColumnFinderFeature: true,
        isGridPagerSimpleMode: true,
        fullGridControllerActionPath: HttpContext.Request.Url.AbsolutePath,
        gridActionName: string.Empty,
        drillThroughControllerAction: string.Empty,
        saveGridControllerAction: string.Empty,
        exportToExcelGridControllerAction: string.Empty
    );
```

```
20
21      // Set dReveal Data Strucuture
22      viewModel.SetState(gridState);
23      viewModel.LoadInitialConfiguration();
24
25      return View("GridReport", viewModel);
26  }
```

View

The dReveal column selection component provides a method to be able of visualizing it throughout a razor page. It is possible to invoke this method following manner:

```
@Html.InfoArch().RenderColumnsSelection(Model.ColumnSelection);
```

The `ColumnSelectionSettings` is the parameter required by the method mentioned above, and prepared in the controller to load it in the view.The following code shows how to use the method in a view:

```
1  <div id="column-selection-component">
2      @Html.InfoArch().RenderColumnsSelection(Model.ColumnSelection);
3      <button id="apply-button" type="button">Apply</button>
4  </div>
```

> **NOTE**
>
> These HTML buttons must be placed within the form tag.

It is important to mention that it is needed to implement the click events for the **Clear** and **Apply** HTML buttons. Additionally, the dReveal column selection component provides a couple of methods on the client side. These methods are used to apply the column selection values selected, and the other one to clear them.

Apply selected columns

The method to apply selected columns is used to send the selected columns information from the client side to the controller action method in the server side described previously. The apply

method received one parameter, this parameter is the reference of the element inside of the form tag.

The following code demonstrates how to implement this method:

```
dRevealColumnSelection.send(element);
```

Example

The following code is a full sample of dReveal column selection component integration:

```
@using InfoArch.Web.Mvc.UI
@using InfoArch.Web.Mvc.Grid
@using InfoArch.Web.Mvc.Enums

@model GridViewModel
@Html.InfoArch().RegisterRequiredThirdPartyLibraries(
    RegisterThirdPartyType.Grid,
    RegisterThirdPartyElements.JQuery
)
@Html.InfoArch().GetStyleSheets(
    new StyleSheet { ExtensionType = ExtensionType.Grid }
)
@Html.InfoArch().GetScripts(
    new Script { ExtensionSuite = ExtensionSuite.Grid }
)

<style>
    #grid-container {
        height: 100vh;
        width: 80%;
    }

    #column-selection-container {
        width: 20%;
        height: 100vh;
    }

    #reporting-grid {
        display: flex;
        flex-direction: row;
    }
</style>
```

```
33
34  <div>
35      @using (@Html.BeginForm("GridReport", "Reporting", FormMethod.Post, new
    { id = "grid-form" }))
36      {
37          @Html.AntiForgeryToken()
38          <div id="reporting-grid">
39              <section id="column-selection-container">
40                  <div id="column-selection-component">
41
    @Html.InfoArch().RenderColumnsSelection(Model.ColumnSelection);
42                      <button id="apply-button" type="button">Apply</button>
43                  </div>
44              </section>
45              <section id="grid-container">
46                  @Html.InfoArch().Grid(@Model.GridSettings).GetHtml()
47              </section>
48          </div>
49      }
50  </div>
51  <script>
52      document.addEventListener("DOMContentLoaded", function () {
53          let applyButton = document.getElementById("apply-button");
54
55          applyButton.addEventListener("click", (event) => {
56              dRevealColumnSelection.send(applyButton);
57          });
58      });
59  </script>
```

> It is important to take in account that the column selection component is into a form tag **<form>**.

> **TIP**
>
> **Security Tip:** The @Html.AntiForgeryToken() method could be added to prevent CSRF attacks. This method is provided by .Net Framework. For further information, visit the official Microsoft documentation.

> **TIP**