Admin Layout and Customization

Smileys

😁 ❤ 🔍 ☑ 💻 📽 📰 ⚖ 🛵 👁 🎮 ⚙

Math

Area of a circle is πr^2

And, the area of a triangle is:

$$A_{triangle} = \frac{1}{2}(b \cdot h)$$

Admin Layout

When the package is installed, a default admin layout is applied on all of the admin pages. Default Admin Layout now support Sidebar and Top Navigation, but only one can be enabled at a time. Default Admin Layout has the following configurations in the sampletoGlobalConfig table:

AdminLayout: This key is used to set the Default Layout page for Admin pages. The default value is:

~/Areas/Admin/Views/Layouts/_AdminLayout.cshtml

AdminLogo: This key is used to set the Logo on the Admin pages. The default value is:

/Areas/Admin/Content/Images/sample_logo.png

AdminName: This key is used to set the App Name on the Admin pages. The default value is:

sampleto Admin Panel

EnableAdminSideBar: This key is used to enable/disable Sidebar on Admin on the Admin pages. The default value is:

true

Custom Admin Layout

There is an option to change the Layout for Admin pages and design it according to your Web Application. There are few things which you first need to set.

NOTE: The admin layout body is designed in Bootstrap 3. So, you must design your custom admin layout in Bootstrap 3.

Changing Layout Page

To change the Layout page, you need to change the default Admin Layout. Change the value for Global Config: 'AdminLayout'. Set its value to the path of your new layout page.

Admin Bundles

You should add the following CSS and Javascript bundles in your admin layout:

- CSS Bundles:

Add the following bundles in the header tag of your Layout page:

```
@Styles.Render("~/bundles/sampletoBootstrap3css")
@Styles.Render("~/bundles/sampletocommoncss")
@Styles.Render("~/bundles/sampletoSitecss")
```

- Javscript Bundles:

Add the following bundles at the end of body tag in your Layout page:

```
@Scripts.Render("~/bundles/sampletocommonscripts")
@Scripts.Render("~/bundles/sampletoBootstrap3js")
@Scripts.Render("~/bundles/sampletoTemplateCommonscripts")
```

For more details, see the Bundles and Optimzation.

Admin Sections

You should add the following CSS and Script sections in your admin layout:

- CSS Section:

Add the following section in the header tag of your Layout page:

```
@RenderSection("css", required: false)
```

- Script Section:

Add the following section at the end of body tag in your Layout page:

```
@RenderSection("scripts", required: false)
```

Getting Logo and App Name

To show the Logo and App Name, you have to get the following Global Configs from sampleto Configuration Provider.

1. Admin Logo and App Name
   - '*AdminLogo*' will contain the Admin App Logo
   - '*AdminName*' will contain the Admin App Name
2. Client Logo and App Name
   - '*ClientLogo*' will contain the Client Web App Logo
   - '*ClientName*' will contain the Client Web App Name

Example:

```
<a class="navbar-brand" href="/">
        @{
            var logo = sampleto.Framework.Core.Config.ConfigurationProvider.GetConfig("Admi

            if (string.IsNullOrEmpty(logo))
            {
                logo = "/Areas/Admin/Content/sampleto/Images/sample_logo.png";
            }

            <img onerror="this.onerror = null; this.src='/Areas/Admin/Content/sampleto/Imag

            var clientName = sampleto.Framework.Core.Config.ConfigurationProvider.GetConfig

            if (string.IsNullOrEmpty(clientName))
            {
                clientName = "sampleto Framework Web";
            }
            <span>@clientName</span>
        }

    </a>
```

Getting Navigations

- Default Navigation View:

You can use the default navigations or create your own view for navigations. In order to use the default view, you must add the following line of code:

```
@Html.Action("GetMenuParital", "Menu", new { menuTypeId = "PRIMARY", viewName = "~/Areas/Ad
```

- Custom Navigation View:

In order to create your own view for navigation, you must follow these instructions:

1. Create a Partial View and strongly bind it with IEnumerable of 'MenuItemvm':

```
@model IEnumerable<sampleto.Framework.ViewModels.Navigation.MenuItemVm>
```

2. Design your Navigtion as per your design requirements.
3. Use the following line of code inorder to show display your navigations:

```
@Html.Action("GetMenuParital", "Menu", new { menuTypeId = "PRIMARY", viewName = [Path_To_Yo
```

Global Hidden Variables

There are few global variables which are needed on every admin page. These variables contain the Multilingual Texts in the form of the hidden field. These contain confirmation alert texts, buttons texts, etc. Add the following line of code at the start of body tag in your Layout page.

```
@Html.Partial("~/Areas/Admin/Views/Shared/AdminLayout/_top_global_vars.cshtml")
```

Following variables are included in the above partial view:

Input	Usage (C#)
`<input type="hidden" id="NoneText" value="@Global.NoneText" />`	Global.NoneText
`<input type="hidden" id="NoDataFoundText" value="@Global.NoDataFoundText" />`	Global.NoDataFoundText
`<input type="hidden" id="ActivateText" value="@Global.ActivateText" />`	Global.ActivateText
`<input type="hidden" id="DeactivateText" value="@Global.DeactivateText" />`	Global.DeactivateText
`<input type="hidden" id="ActiveText" value="@Global.ActiveText" />`	Global.ActiveText
`<input type="hidden" id="InactiveText" value="@Global.InActiveText" />`	Global.InActiveText
`<input type="hidden" id="EditText" value="@Global.EditText" />`	Global.EditText
`<input type="hidden" id="DeleteText" value="@Global.DeleteText" />`	Global.DeleteText
`<input type="hidden" id="ConfirmDeleteRecordText" value="@Global.ConfirmDeleteRecordText" />`	Global.ConfirmDeleteRecordText

Input	Usage (C#)
`<input type="hidden" id="ConfirmDeactivateRecordText" value="@Global.ConfirmDeactivateRecordText" />`	Global.ConfirmDeactivateRecordTe
`<input type="hidden" id="ConfirmActivateRecordText" value="@Global.ConfirmActivateRecordText" />`	Global.ConfirmActivateRecordText
`<input type="hidden" id="OperationFailedText" value="@Messages.OperationFailedText" />`	Messages.OperationFailedText
`<input type="hidden" id="OperationPerformedSuccessfullyText" value="@Messages.OperationPerformedSuccessfullyText" />`	Messages.OperationPerformedSuc
`<input type="hidden" id="ConfirmClearLoginHistoryText" value="@Messages.ConfirmClearLoginHistoryText" />`	Messages.ConfirmClearLoginHistor
`<input type="hidden" id="MissingRequiredFields" value="@Messages.MissingRequiredFields" />`	Messages.MissingRequiredFields

Azure Devops Work Item link

#7

- Working link in list
- #8

Working link between #7 text

#6 - A bug

#NotAWorkItemId

Non working link#7 #7Non working link Definitly#7Non working link