Contents

[How to create a Report List 2](#_Toc8648461)

[Naming Convention 2](#_Toc8648462)

[Location in Project 2](#_Toc8648463)

[Creating a Class for Report 4](#_Toc8648464)

[Base Attributes 5](#_Toc8648465)

[Common 5](#_Toc8648466)

[Controls 5](#_Toc8648467)

[Access 6](#_Toc8648468)

[Creating a View for Report 6](#_Toc8648469)

[Front End Report List 7](#_Toc8648470)

# How to create a Report List

Report List is similar to a table report with rows and columns of data. Each row is one record and each column is a field. This type of report is often used when you want to see more number of records at a time for preview, sort or search for record.

To create Report List need to create :

1. Add Class
2. Add View

# Naming Convention

For C# class names and corresponding table names are used only English singular terminology in Camel Case (<https://ru.wikipedia.org/wiki/CamelCase>). The same approach is used for properties and table columns.

For example, if it is required in Report to add a Student state field with name StudentState. Should be used following names:

Class file: **StudentList**.cs

Class name: **StudentState**

View file: **StudentList**.sql

string Name

Class name should be the same like View name.

# Location in Project

Report class files usually are placed in folder :

1. FrontEnd Reports

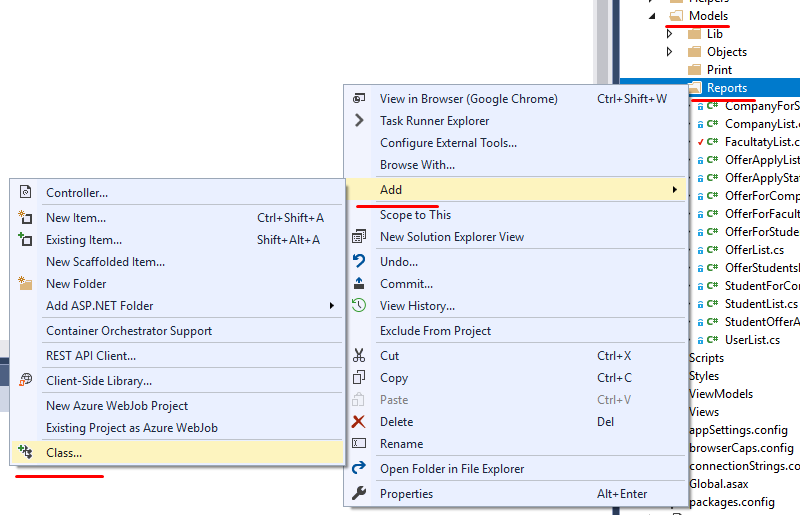
\mvc\Models\Report

View class files usually are placed in folder :

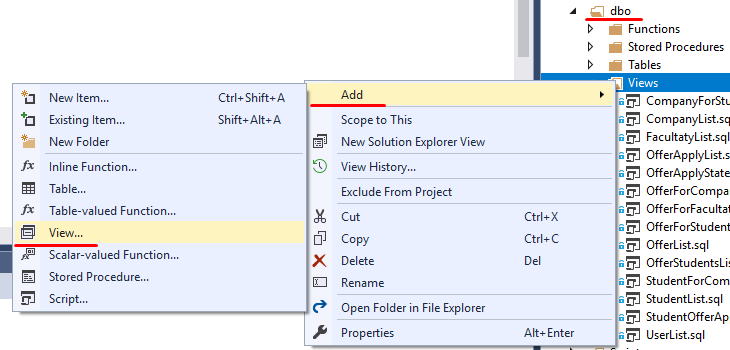
1. FrontEnd Views

Database\dbo\Views

1. Add Class for Report



1. Add SQL View for Report



# Creating a Class for Report

Information

Each Class should have standard Header with basic Information (for example):

// --------------------------------------------------------------------------------------// <copyright file="StudentList.cs" company="GALEXSTUDIO">

// Copyright © 2019

// </copyright>

// <summary>

// The StudentList is a test class with all possible features and controls.

// </summary>

// --------------------------------------------------------------------------------------

Usings

Minimum Using:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using LIB.Tools.Utils;

using LIB.AdvancedProperties;

using LIB.BusinessObjects;

using LIB.Tools.BO;

using LIB.Tools.AdminArea;

using System.Collections.Generic;

Class Header Attributes

Name of Report List which will be displayed on FrontEnd

[Bo(DisplayName = “StudentList” )]

Class Structure

1. Class in which will be placed report columns (properties)

Regions are not obligatory, but is a good style to add it.

public class StudentList : ReportBase

{

#region Constructors

public StudentList()

: base()

{

}

#endregion

2. Link for Report records

public override string GetLink()

{

return "DocControl/Student/" + StudentId.ToString();

}

3. Properties

#region Properties

public long FacultatyId { get; set; }

[Common(DisplayName = "State Name", \_Searchable = true, \_Sortable = true), Template(Mode = Template.Name), Access(DisplayMode = LIB.AdvancedProperties.DisplayMode.Simple | LIB.AdvancedProperties.DisplayMode.Advanced | LIB.AdvancedProperties.DisplayMode.Search | LIB.AdvancedProperties.DisplayMode.Print)]

public string Name { get; set; }

#endregion

}

Class Properties

## Base Attributes

#GoFra framework have background logic for each class that is inherited from ItemBase, that allows objet to be Created, Populated, Canceled, Deleted, Updated In Database. Generated class from Post and Display class using on the controls in UI. All of this logic is based on predefined attributes of class and properties:

### Common

Set of attributes that are responsible for Common settings and display logic.

DisplayName - Common(DisplayName = "State Name") – By default in CP and SMI will be shown property name (English name in CamelCase), if it is needed to show friendly name it should be changed Display Name also the name of Report.

\_Searchable = true - possibility to make Search by records in column (true by default ).

\_Sortable = true - possibility to Sort column (false by default ).

### Controls

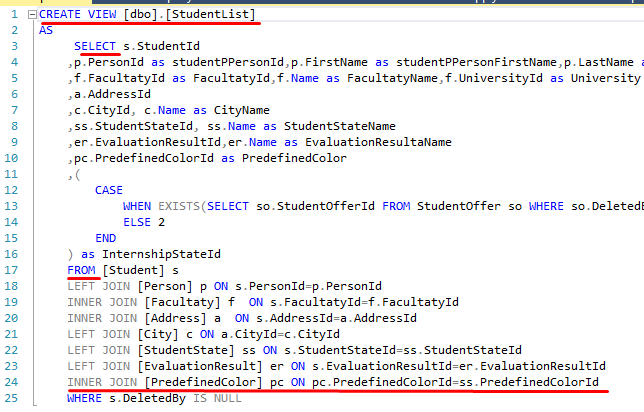
In order to make process of defining properties more simple there were generated a set of templates (for exemple Name, String, Description, Number, Decimal, DropDown, Date etc.):

Template - Template(Mode = Template.Name)

### Access

DisplayMode- Enum that controls in what mode (Datagrid, ItemEdit, Filters, etc.) to show this property. Ex. DisplayMode = DisplayMode.Simple | DisplayMode.FrontEnd | DisplayMode.Advanced | DisplayMode.Print, that will make this property to be presented in datagrid, Item Layout, Filters and on Print.

# Creating a View for Report



1. Name of View

CREATE VIEW [dbo].[StudentList]

2. Select Database columns for View

SELECT s.StudentId,p.PersonId as studentPPersonId

,ss.StudentStateId, ss.Name as StudentStateName

3. Select from table

FROM [Student] s

4. Joins with other Database tables

For example : to highlight StudentState by different color

INNER JOIN [PredefinedColor] pc ON pc.PredefinedColorId=ss.PredefinedColorId

# Front End Report List



1. List name [Bo(DisplayName = “StudentList” )]

2. Set column for Preview or Print

3. Filters Access(DisplayMode = LIB.AdvancedProperties.DisplayMode.Search)

4. Column name (DisplayName = "State Name")

5. Report records