Build Form Layout

Contents

[1. Introduction 2](#_Toc13579393)

[2. Add Form 3](#_Toc13579394)

[2.1 Class For Form 3](#_Toc13579395)

[2.2 View for Form 4](#_Toc13579396)

[2.3 Add list for form 5](#_Toc13579397)

[2.3.1 Add list in the View of the Class 6](#_Toc13579398)

[2.3.2 Partial view of list 6](#_Toc13579399)

[2.3.3 Default List in Form 10](#_Toc13579400)

[3. Form Rows 11](#_Toc13579401)

[4. Form Templates 12](#_Toc13579402)

[5. Form Controller 13](#_Toc13579403)

1. Introduction

The form represent a page where is situated the form. It is consisted of fields of a certain entity, filling it will be created an entry in the database.

The form attributes represent the components that make up a form. They are:

* Templates
* Rows
* Controls

\* Replace Test by your class

# 2. Add Form

## 2.1 Class For Form

1. Add object to the Models\Objects. This object should inherit from ModelBase or PrintBase

public class Test: PrintBase

1. All properties that will be used in form should have attributes as described in the Document “Add Class”

[Common(Order = 1, DisplayName = "Name"), Template(Mode = Template.Name), Validation(ValidationType = ValidationTypes.Required)]

\* Informing point (This methods will be used in future, so do not accent your attention on it)

Exist several methods, used in fact automatic, but they also can be overridden for changing the logic of the form, adding specific fields, saving the form in more than one table, or other nonstandard actions.

They are: PopulateFrontEnd()

SaveForm() //Saving the values from form fields in the db table

CollectFromForm() //collecting the values from the fields in the form

PopulateFrontEndItems()

PopulateOne()

(…And others)

## 2.2 View for Form

1. Add View at Views\ DocControl\Test.cshtml

View Name should be the same like class name.   
 Ex: Test.cshtml.

1. It is possible to get to the form from Dashboard (by click on icon, look Document AddIconOnDashBoard), by click on list item in Reports or DashBoard List, or other ways.

@model MedCore.Models.Objects.Test

@using Weblib.Models.Common

@using LIB.BusinessObjects

@using Person = MedCoreLib.BusinessObjects.Person

@{

Layout = "~/Views/Master/\_Standard.cshtml";

var currentUser = LIB.Tools.Security.Authentication.GetCurrentUser();

}

<form class="inner-content-area content-control content-two-columns" action="@LIB.Tools.Utils.URLHelper.GetUrl("DocControl/Save")">

<div class="ajax-loading-overlay"></div>

@{

List<ButtonModel> ControlButtons = new List<ButtonModel>();

if (Model.Id > 0 && currentUser.HasAtLeastOnePermission((long)LIB.BusinessObjects.BasePermissionenum.SuperAdmin))

{

ControlButtons.Add(new ButtonModel() { Name = "Edit", Text = "Redactarea Avansata", Href = LIB.Tools.Utils.URLHelper.GetUrl("ControlPanel/EditItem/Test/GofraVersions.Models.Objects/" + Model.Id.ToString()) });

}

}

@Html.Partial("../Controls/RowControl/\_pagecontrols", new PageControlsModel()

{

Id = Model.Id,

Buttons = ControlButtons,

Delete = true,

Object = "Test",

Save = true,

Namespace = typeof(MedCore.Models.Objects.Test).FullName

})

@Html.AntiForgeryToken()

@Html.ValidationSummary(true)

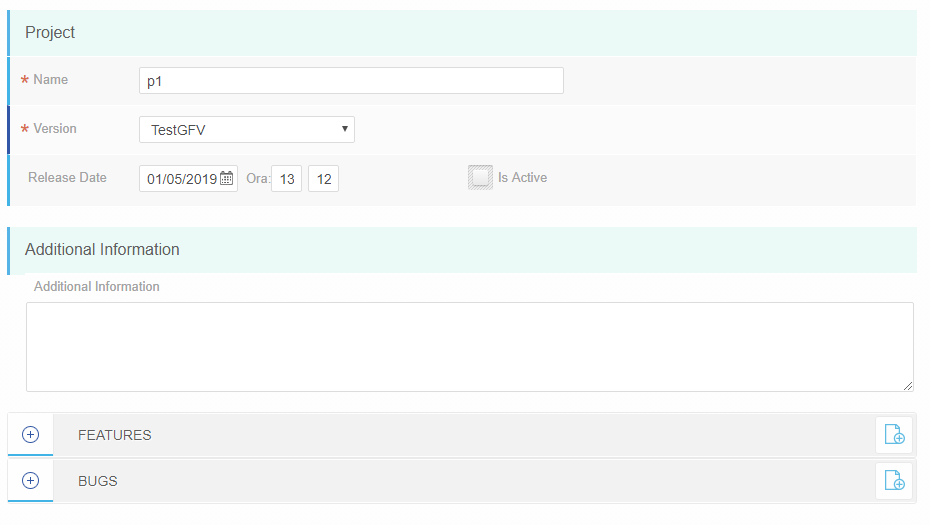
//else…

</form>

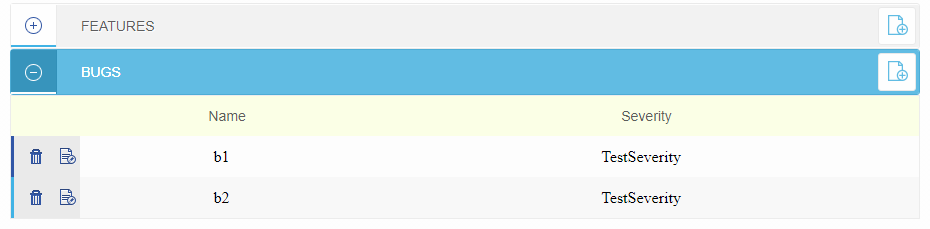
## 2.3 Add list for form

This kind of list is used in forms to add objects from other classes to this class.

It is possible to add some object by click on button.



If you click on  the list will unfold, and will appear the list of added objects. They can be deleted  or updated.



\* Rename the names to the necessary in your project

### 2.3.1 Add list in the View of the Class

Add this part of code in the end of the form.

<div class="edit-section dynamic-section dynamic-section-autosave">

<div class="edit-section-header">

<a href="#" id="dynamic\_section\_notes" onclick="return toggle\_dynamic\_section(this, '@typeof(TestType).FullName', '@typeof(TestType).Name', false)" title="Extinde" class="edit-section-state-indicator"></a>

<div class="edit-section-header-caption"> TestTypes</div>

<a href="@(LIB.Tools.Utils.URLHelper.GetUrl("DocControl/Test") + "/0" )" onclick="@LIB.Tools.Utils.URLHelper.GetUrl("DocControl/Test")" title="Add new" class="edit-section-icons edit-section-new"></a>

<div class="clear"></div>

</div>

<div class="edit-section-body">

</div>

<div class="section-controls" style="display: none">

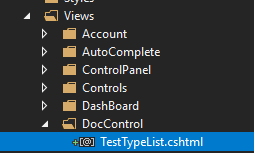
@Html.Partial("../Controls/\_button", new ButtonModel() { Name = "Save", Text = "Save", Action = "Save(" + Model.Id + ", this)" })

</div>

</div>

### 2.3.2 Partial view of list

1. Create a view for list in your project by the example:



Add the following code into this view:

@model Dictionary<long, LIB.Tools.BO.ItemBase>

@using System.IdentityModel.Protocols.WSTrust

@using MedCore.Models.Objects

@using LIB.AdvancedProperties

@using Weblib.Models

@using Weblib.Models.Common

@{

Layout = "~/Views/Master/\_Ajax.cshtml";

var currentUser = LIB.Tools.Security.Authentication.GetCurrentUser();

var properties = (LIB.AdvancedProperties.AdvancedProperties)ViewData["properties"];

}

<div class="edit-section-table">

<input type="hidden" id="dNamespace" value="@ViewData["item"].GetType().FullName" />

@\*The names for columns\*@

<div class="edit-section-table-header">

<div class="table-title">

</div>

<div class="edit-section-table-title">

Name

</div>

<div class="edit-section-table-title">

Something

</div>

</div>

@\*The rows of the list and the buttons of deleting and editing the rows\*@

@if (Model != null)

{

foreach (var item in Model.Values)

{

var test = (TestType)item;

<div class="edit-section-table-row">

<input type="hidden" name="DynamicObjectId" value="@(test.Id)" />

<div class="edit-section-table-content edit-section-table-content-controls edit-section-table-content-controls-fixed">

<a title="Delete" class="control-delete" onclick="return DeleteTestTypeFromTest(@testType.Id, this)" href="#"></a>

<a title="Change" class="control-edit" id="change" onclick="" href="@(LIB.Tools.Utils.URLHelper.GetUrl("DocControl/TestType") + "/TestTypeId=" + testType.Id)"></a>

</div>

<div class="edit-section-table-content">

@test.GetName()

</div>

<div class="edit-section-table-content">

@(test.Something!= null ? test.Something.GetName() : "")

</div>

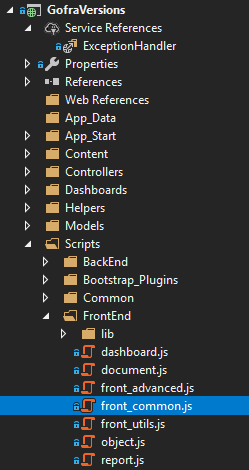
</div>

}

}

</div>

2. Add the script to the file for deleting the row:



function DeleteTestTypeFromTest(testTypeId, element) {

if (confirm('Are you sure?')) {

var eObj = $(element);

var gpostArray = {};

gpostArray["TestTypeId"] = testTypeId;

gpostArray["TestId"] = $('[name="Id"]').val();

$.ajax({

type: 'POST',

url: gRootUrl + "Helper/DeleteTestTypeFromTest/",

async: true,

data: gpostArray,

success: function (response) {

if (response.Result == 1) {

eObj.closest('.edit-section-table-row').remove();

}

else {

console.log(response);

alert(response.Message);

}

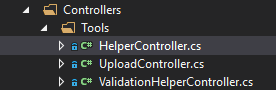
}

});

}

}

3. Add method to controller HelperController.cs (add this controller if it is necessary)



using MedCore.Models.Objects;

using LIB.Tools.Utils;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using LIB.Tools.BO;

using LIB.Helpers;

namespace MedCore.Controllers.Tools

{

public class HelperController : Controller

{

public ActionResult DeleteTestTypeFromTest(long TestTypeId, long TestId)

{

try

{

const string StrSql = "TestType\_DeleteFromTest";

var conn = DataBase.ConnectionFromContext();

var cmd = new SqlCommand(StrSql, conn) { CommandType = CommandType.StoredProcedure };

var param1 = new SqlParameter("@TestId", SqlDbType.BigInt) { Value = TestId };

var param2 = new SqlParameter("@TestTypeId", SqlDbType.BigInt) { Value = TestTypeId };

cmd.Parameters.Add(param1);

cmd.Parameters.Add(param2);

cmd.ExecuteNonQuery();

}

catch (Exception e)

{

return Json(new RequestResult { Result = RequestResultType.Fail, Message = "Oh No..." });

}

return Json(new RequestResult { Result = RequestResultType.Success, Message = "Oh Yeah..." });

}

}

}

### 2.3.3 Default List in Form

The difference between the custom list and the default list is that there is no need to create a **partial view** (see par.3), since it is generated automatically.

So it is necessary just the code in the main view:

<div class="edit-section dynamic-section">

<div class="edit-section-header">

<a href="#" onclick="return toggle\_dynamic\_section(this, 'MedCoreLib.BusinessObjects.Test', 'Test',false,'','Id')" title="Extinde" class="edit-section-state-indicator"></a>

<div class="edit-section-header-caption"> Test </div>

<a target="\_blank" href="@LIB.Tools.Utils.URLHelper.GetUrl("/PuthYouNeed/"+Model.Id)" title="Add Test" class="edit-section-icons edit-section-new"></a>

<div class="clear"></div>

</div>

<div class="edit-section-body">

</div>

</div>

# 3. Form Rows

For two column rows

<div class="edit-section-row">

<div class="edit-section-row-left">

</div>

<div class="edit-section-row-right">

</div>

<div class="clear"></div>

</div>

For one full-size row with float elements

<div class="edit-section-row edit-section-row-full-size-floatelements">

<div class="clear"></div>

</div>

For three column rows

<div class="edit-section-row">

<div class="edit-section-row-trio edit-section-row-fisrt">

</div>

<div class="edit-section-row-trio edit-section-row-second">

</div>

<div class="edit-section-row-trio edit-section-row-third">

</div>

<div class="clear"></div>

</div>

<div class="edit-section-row edit-section-row-trio">

<div class="clear"></div>

</div>

For one full-size row

<div class="edit-section-row edit-section-row-full-size">

<div class="clear"></div>

</div>

# 4. Form Templates

For 1 column section

<div class="content-one-column">

<div class="edit-section">

<div class="edit-section-header">

SECTIONNAME

</div>

<div class="edit-section-body">

</div>

</div>

</div>

<div class="clear separator"></div>

For 2 column sections

<div class="inner-content-left">

<div class="edit-section">

<div class="edit-section-header">

SECTIONNAME

</div>

<div class="edit-section-body"></div>

</div>

</div>

<div class="inner-content-right">

<div class="edit-section">

<div class="edit-section-header">

SECTIONNAME

</div>

<div class="edit-section-body"></div>

</div>

</div>

<div class="clear separator"></div>

# 5. Form Controller

1. Date

@Html.Partial("../Controls/RowControl/\_inputrow", new TextboxModel()

{

Name = "LaboratoryDate",

Class = "input calendar-input",

Type = Weblib.Models.Common.Enums.TextboxType.DateTime,

Value = Model.LaboratoryDate != DateTime.MinValue ? Model.LaboratoryDate.ToString(@"dd/MM/yyyy", System.Globalization.CultureInfo.InvariantCulture) : "",

DateValue = Model.LaboratoryDate != DateTime.MinValue ? Model.LaboratoryDate : DateTime.MinValue,

Caption = ""

})

1. DropDown

@Html.Partial("../Controls/RowControl/\_selectrow", new DropDownModel()

{

Name = "Status",

AllowDefault = true,

Value = (Model.Status != null ? Model.Status.Id.ToString() : "0"),

Options = (new Status()).Populate(),

Caption = "",

ValidationType = ValidationTypes.None

})

1. Number

@Html.Partial("../Controls/RowControl/\_inputrow", new TextboxModel()

{

Name = "SamplingTemp",

Type = Weblib.Models.Common.Enums.TextboxType.Number,

Value = Model.SamplingTemp!=int.MinValue?Model.SamplingTemp.ToString():"",

Class = "input number-input",

Caption = ""

})

1. Decimal

@Html.Partial("../Controls/RowControl/\_inputrow", new TextboxModel()

{

Name = "SamplingTemp",

Type = Weblib.Models.Common.Enums.TextboxType.Number,

Value = Model.SamplingTemp!=decimal.MinValue?Model.SamplingTemp.ToString("0.0"):"",

Class = "input number-input",

Caption = ""

})

1. CheckBox

@Html.Partial("../Controls/RowControl/\_checkboxrow", new CheckboxModel()

{

Name = "RequestType",

Checked = (Model.RequestType == null || (Model.RequestType != null && Model.RequestType.Id == 1)),

Caption = ""

})

1. Foto

@Html.Partial("../Controls/RowControl/\_photorow", new PhotoModel()

{

Name = "Foto",

Value = Model.Foto,

Width = 205,

Height = 210

})

<script type="text/javascript">

initUploadImageFile('@LIB.Tools.Utils.URLHelper.GetUrl("Upload/DoUploadImage/")', 'Foto', 62, 41, 615, 630,"Person");

</script>

1. File

@Html.Partial("../Controls/RowControl/\_filerow", new DocumentModel()

{

Name = DevelopmentExecutionPaymentPrefix + "Document",

ReadOnly = bReadonly,

Value = Model.DevelopmentExecutionPaymentFinalized != null ? Model.DevelopmentExecutionPaymentFinalized.Document : null,

Caption = SAPSIMA.Helpers.T.Str("DovadaPlataCopiaScanata", "ObjectFolder", "Copia Scanata")

})