

# SharpShooter Reports.Web Using Web Viewer in MVC Application

---

Last modified on: September 5, 2011



## Table of Content

System Requirements .....	3
Introduction.....	3
Step 1. Creation of a project.....	3
Step 2. Setting up a Web application .....	4
Step 3. Implementation of Service Logic .....	6
Step 4. Adding a controller.....	6
Step 5. Setting up controller which serves as service.....	7
Step 6. Adding JavaScrip files to the project.....	7
Step 7. Adding styles.....	9
Step 8. Adding default skin files .....	10
Step 9. Adding "View" to the project.....	12
Step 10. Adding scripts and styles to the view .....	14
Step 11. Adding the Report Viewer component to the View.....	15
Step 12. Running the application.....	18



## System Requirements

In order to use SharpShooter Reports.WebViewer successfully in ASP.NET MVC Web applications you will need:

- .NET Framework 4
- Visual Studio 2010
- ASP.NET MVC 3

## Introduction

The purpose of this getting started is to cover the basic use and provide minimum required skills necessary for work with SharpShooter Reports.Web. This is step-by-step tutorial for creation of web application with SharpShooter Reports.Web. Let's examine the creation and configuration of a service, creation of a report and integration of report web viewer.

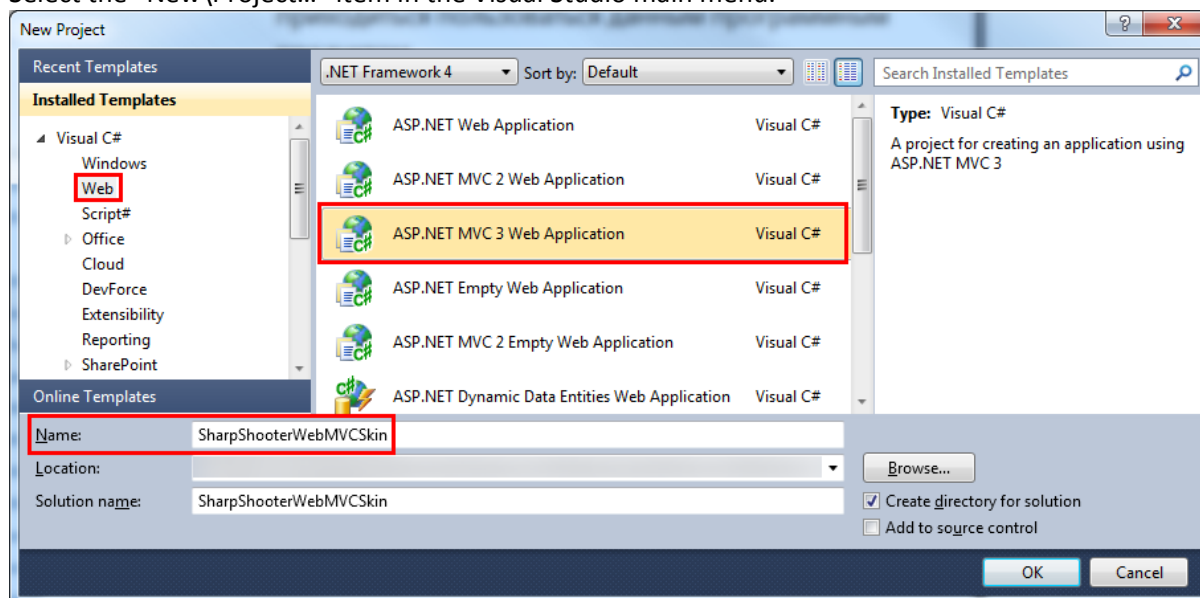
Following this step-by-step tutorial, you will be able to configure a service and add report viewer with predefined skin into your web-application or your website in 15 minutes.

Here, we describe the process for web application creation with use of SharpShooter Reports.Web. We will examine the creation and configuration of a service, report creation and integration of report viewer to a web application page.

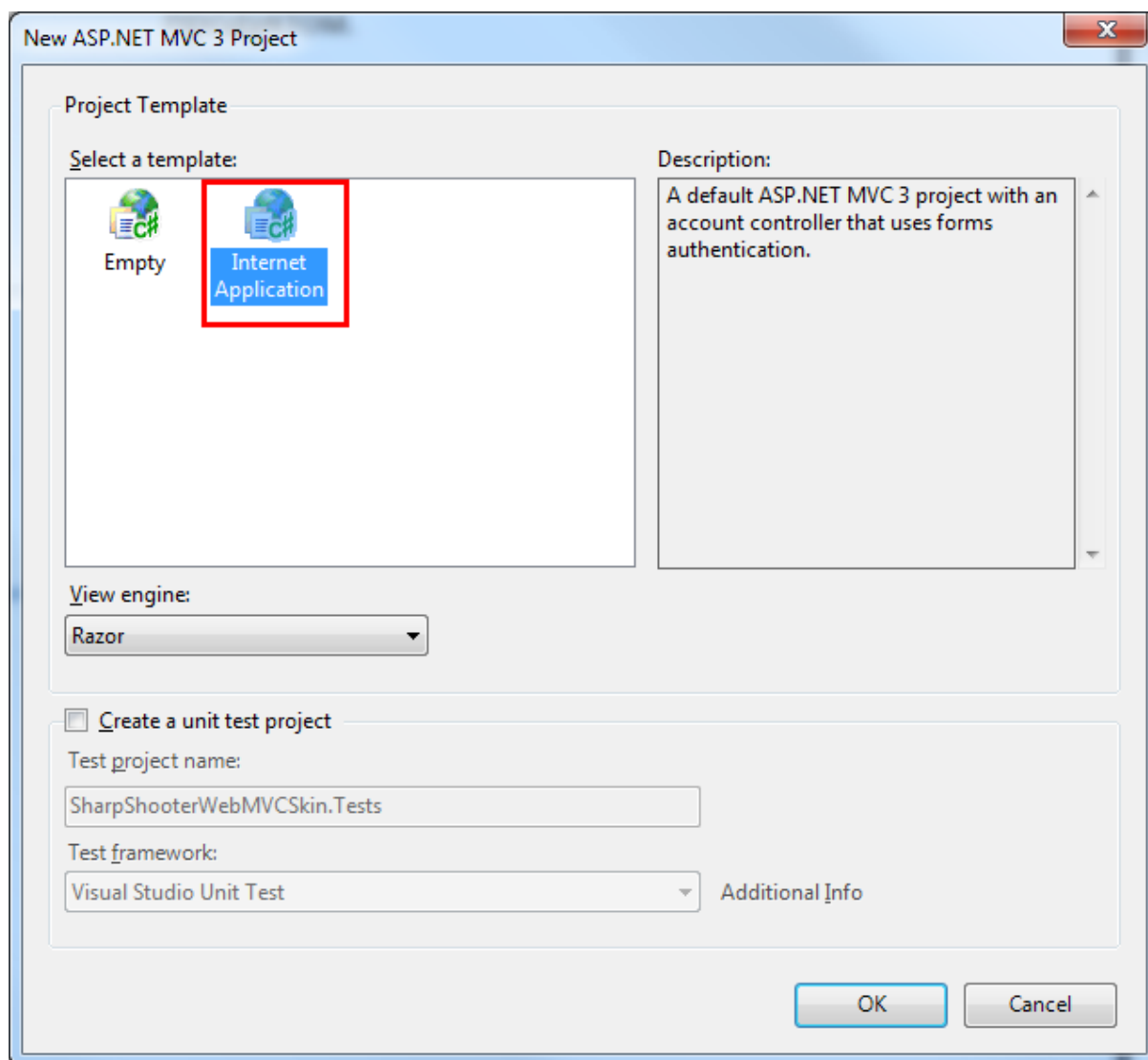
### Step 1. Creation of a project

Create a new "ASP.NET MVC 3 Web Application" project and set a name to "SharpShooterWebMVCSkin".

Select the "New\Project..." item in the Visual Studio main menu.

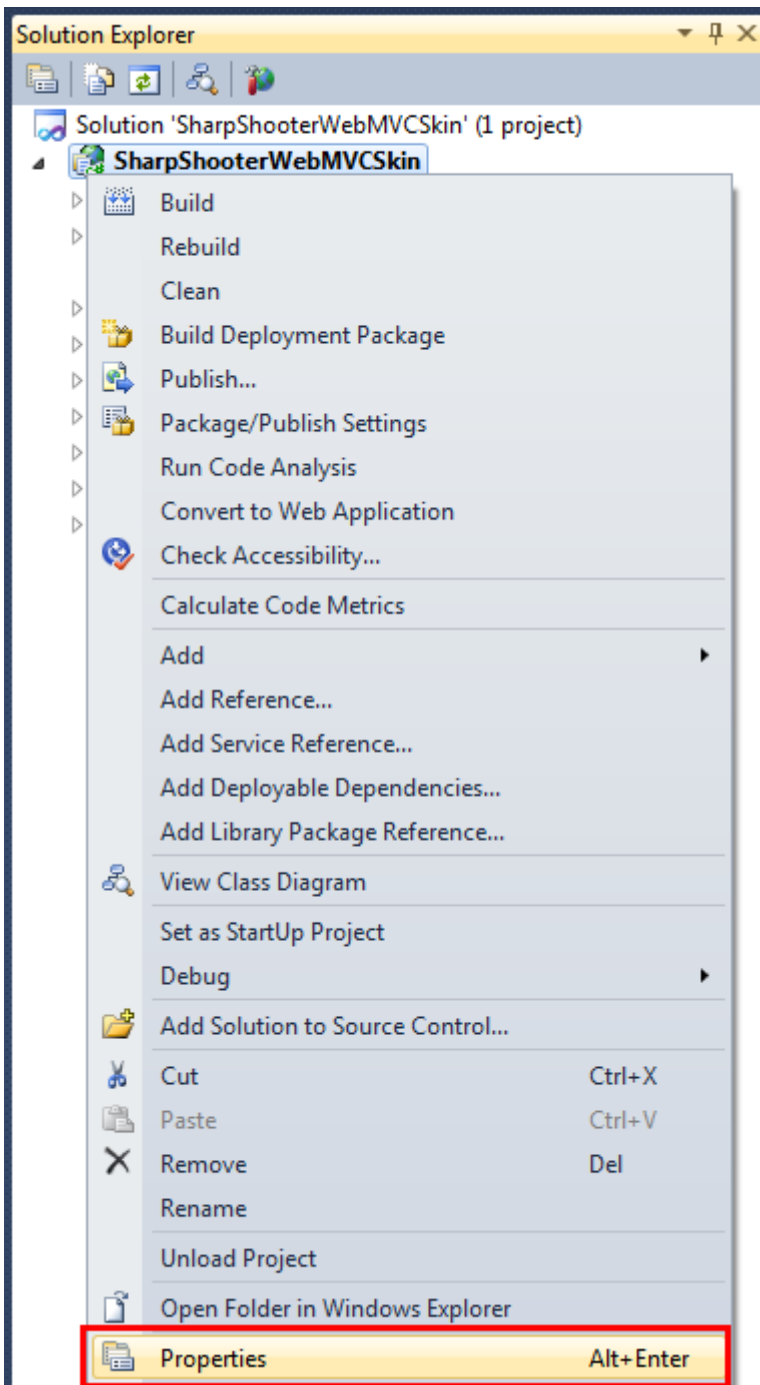


Select the "Internet Application" template and click the "Ok" button.

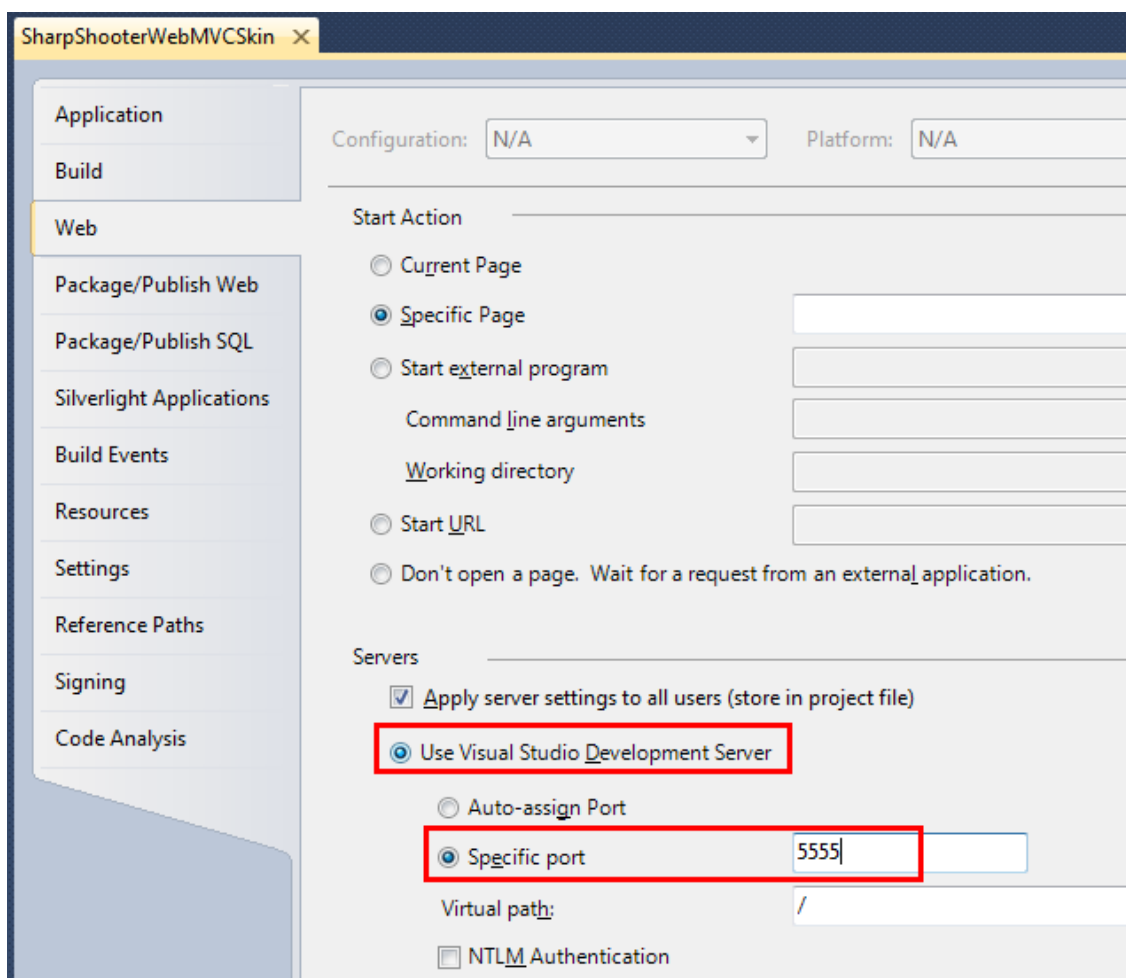


## Step 2. Setting up a Web application

Change the properties of the “SharpShooterWebMVCSkin” project. Invoke the context menu and select the “Properties” item.



Set “Use Visual Studio Development Server”, “Specific port” equal to 5555 in the “Web” tab.

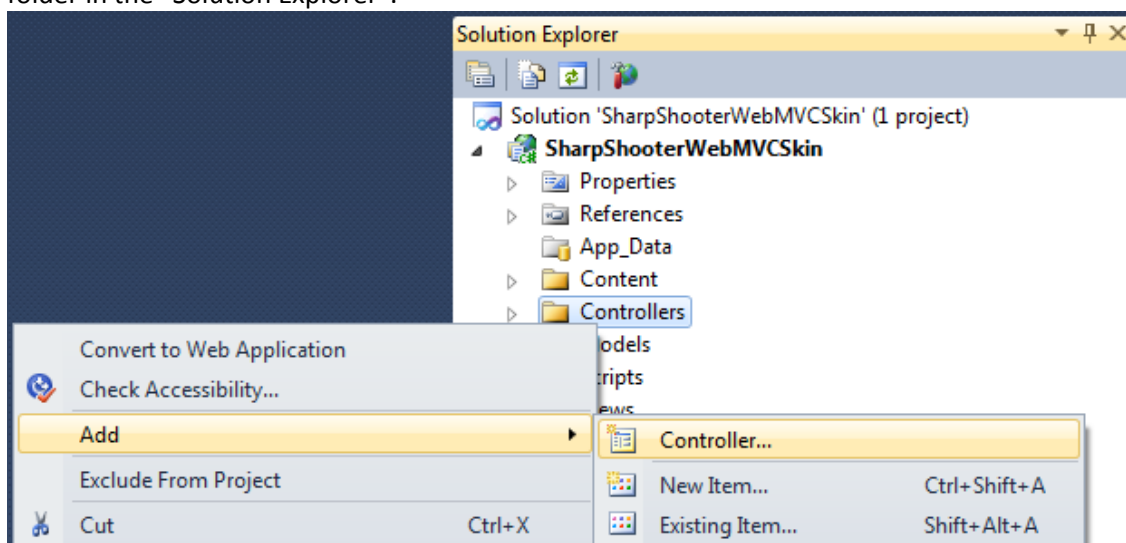


### Step 3. Implementation of Service Logic

Add the class which implements the service for Report Engine to the project. More detailed description on how to add this class is available in [Configuring Reporting Engine](#).

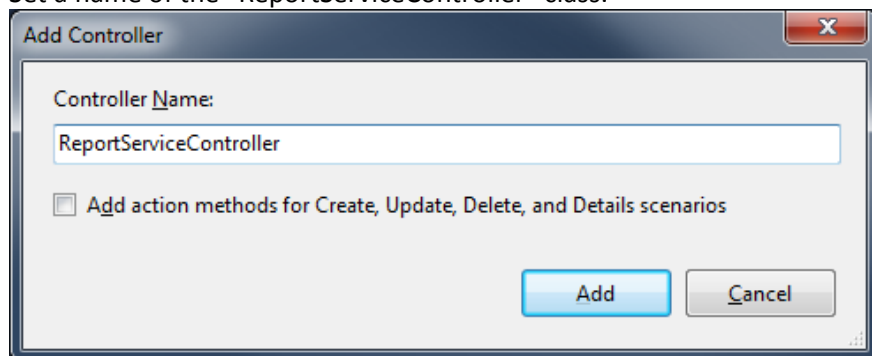
### Step 4. Adding a controller

Add a controller to the project. Select the “Add\New Item...” item in the context menu of the “Controllers” folder in the “Solution Explorer”.





Set a name of the “ReportServiceController” class.



## Step 5. Setting up controller which serves as service

Add the “PerpetuumSoft.Reporting.WebViewer.Server.Mvc” namespace to the controller:

```
using PerpetuumSoft.Reporting.WebViewer.Server.Mvc;
```

Inherit the created controller from ReportServiceBaseController and override the “CreateReportService” function. This overridden function will return the “ServiceClass” class instance. As a result, the code in the “ReportServiceController.cs” file should look as follows:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

using PerpetuumSoft.Reporting.WebViewer.Server.Mvc;

namespace SharpShooterWebMVCSkin.Controllers
{
    public class ReportServiceController : ReportServiceBaseController
    {
        protected override PerpetuumSoft.Reporting.WebViewer.Server.ReportServiceBase CreateReportService()
        {
            return new ServiceClass();
        }
    }
}
```

The setting up the server part of the application is finished. Let’s set the client application.

## Step 6. Adding JavaScript files to the project

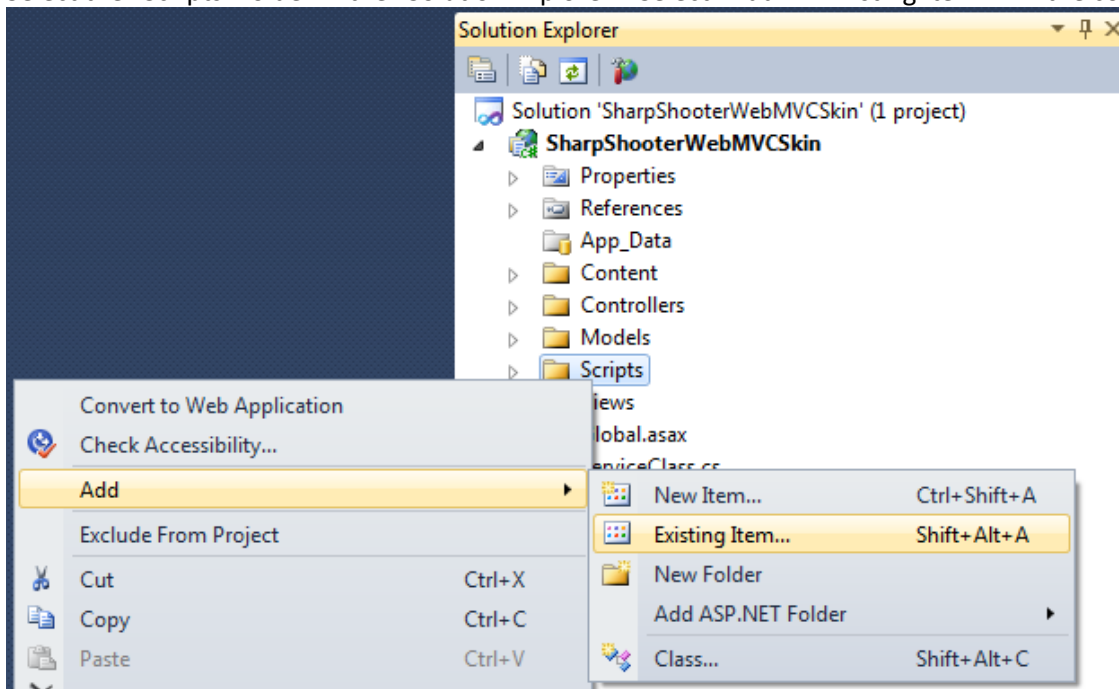
You need to add the following files to the project:

- jquery-1.5.1.js – jQuery plugin;
- jquery-ui-1.8.11.js - jQuery plugin for UI;
- jquery.treeview.js – implements the bookmarks tree functionality;
- mscorlib.js – provides functionality like a type system in script, and basic utility APIs (it’s necessary while using classes creating by Script#);
- PerpetuumSoft.Reporting.WebViewer.Client.js – implements the logic for receiving and showing the reports



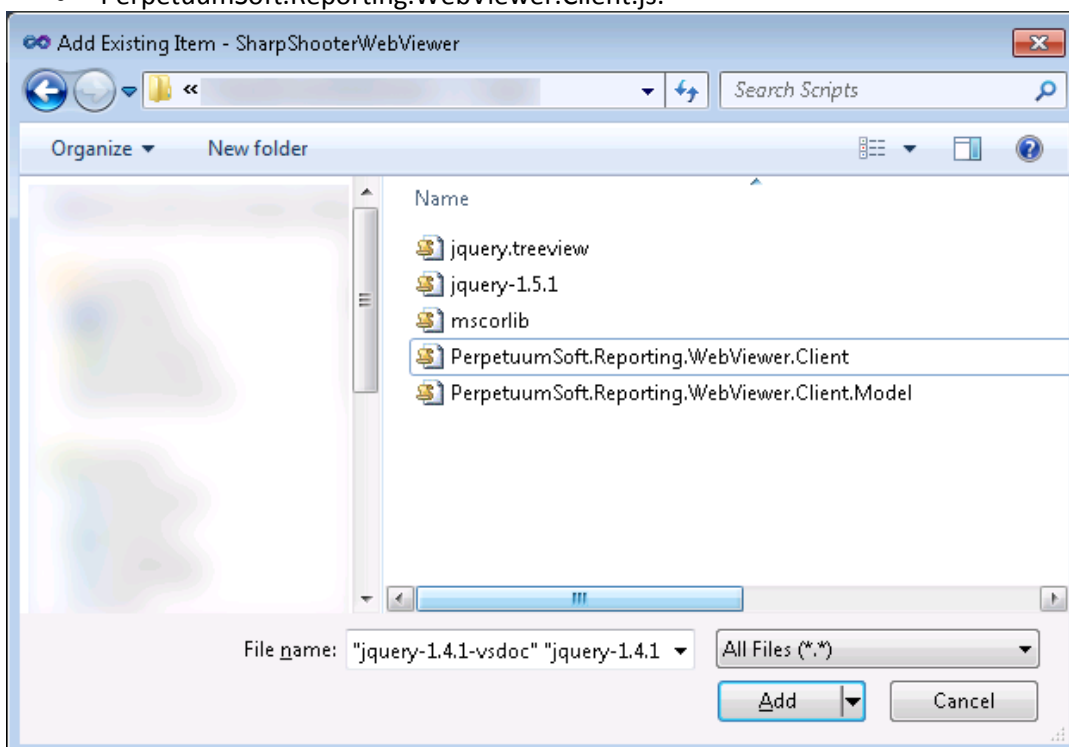
- PerpetuumSoft.Reporting.WebViewer.Client.Model.js – data model classes;
- WebViewer.Style.js – methods controlling the style and skin of the component.

Select the “Scripts” folder in the “Solution Explorer”. Select “Add”->“Existing Item...” in the context menu.



Add the following files from the “Web\Scripts” folder in your installation directory of SharpShooter Reports.Web:

- jquery-1.5.1.js;
- jquery.treeview.js;
- mscorlib.js;
- PerpetuumSoft.Reporting.WebViewer.Client.Model.js;
- PerpetuumSoft.Reporting.WebViewer.Client.js.

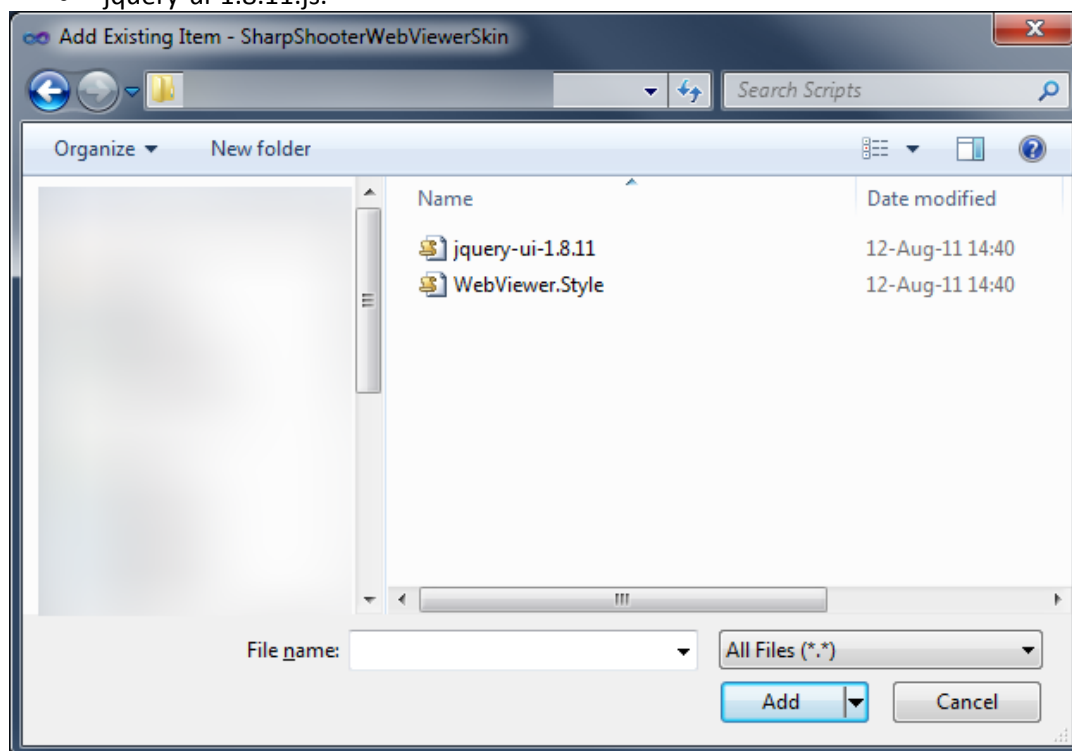






Add the following files from the “Web\Skins\Default\Scripts” folder in your installation directory of SharpShooter Reports.Web:

- WebViewer.Style.js;
- jquery-ui-1.8.11.js.

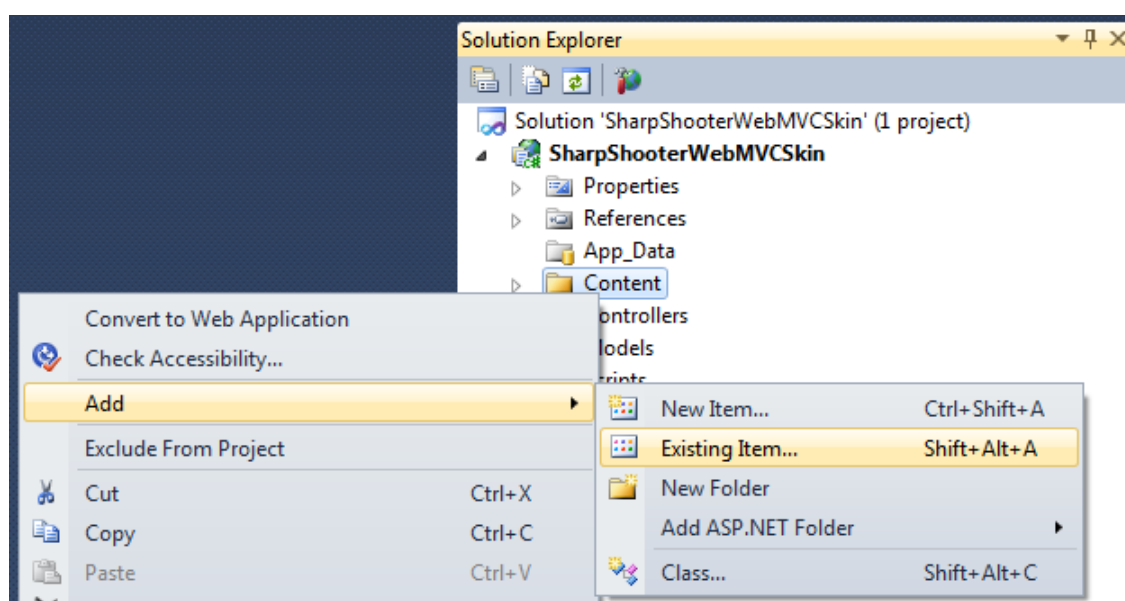


## Step 7. Adding styles

Add the styles to the project. These styles are needed for the correct report display. It is necessary to add the following file with styles to the project:

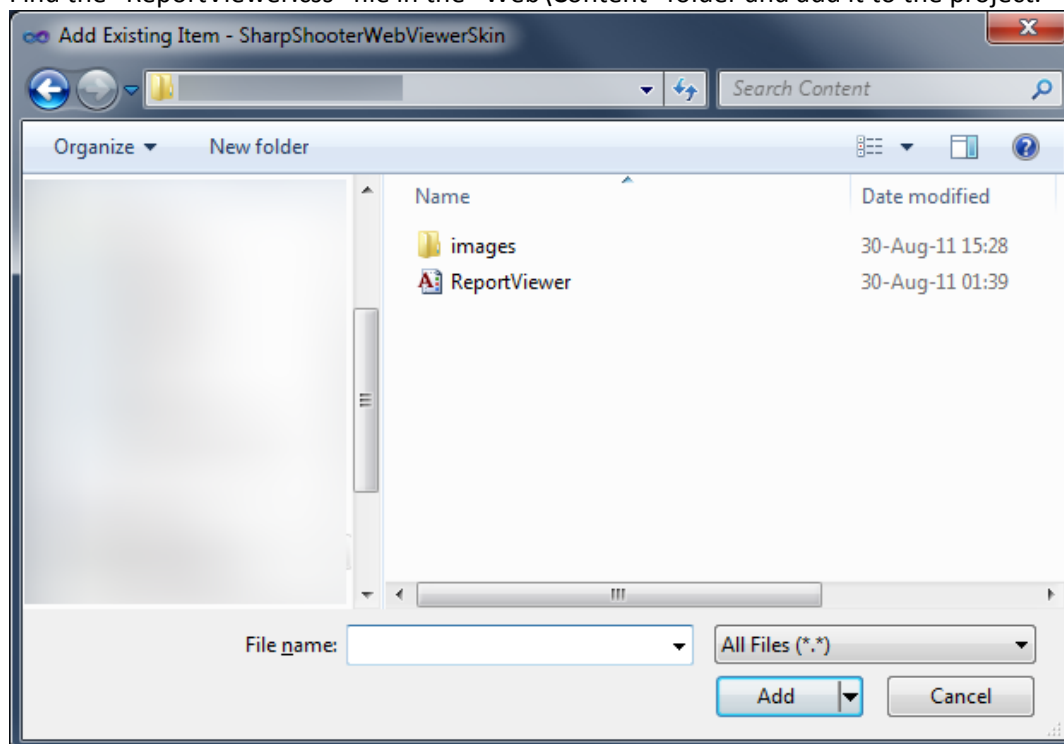
ReportViewer.css – this file describes styles which are used for reports display.

Select the “Content” folder in the “Solution Explorer”. Select “Add”->“Existing Item...” in the context menu.





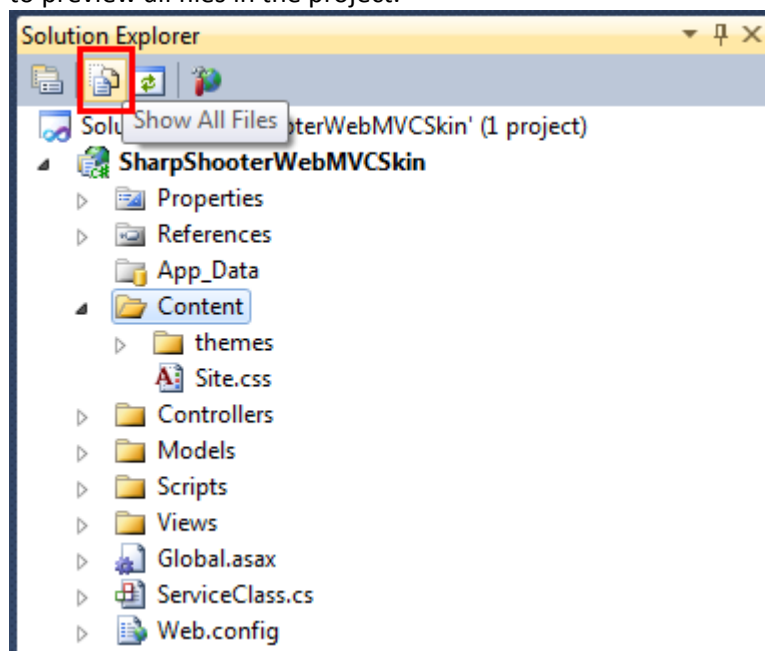
Find the “ReportViewer.css” file in the “Web\Content” folder and add it to the project.



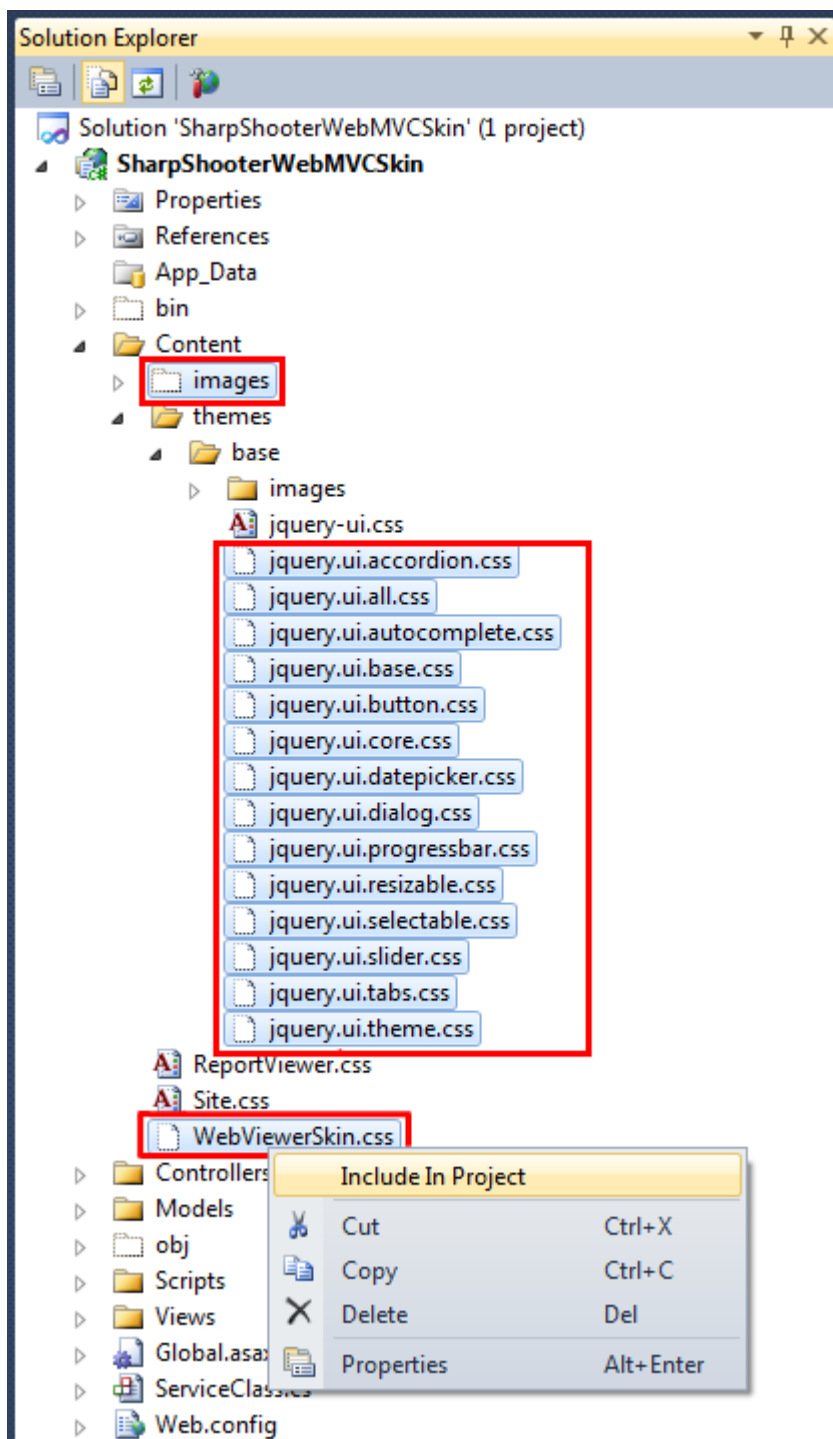
### Step 8. Adding default skin files

Add some files with styles and images. They are used to setup the user interface of the component. Copy the content of the “Web\Skins\Default\Content\” folder in the installation directory to the “Styles” folder of your project.

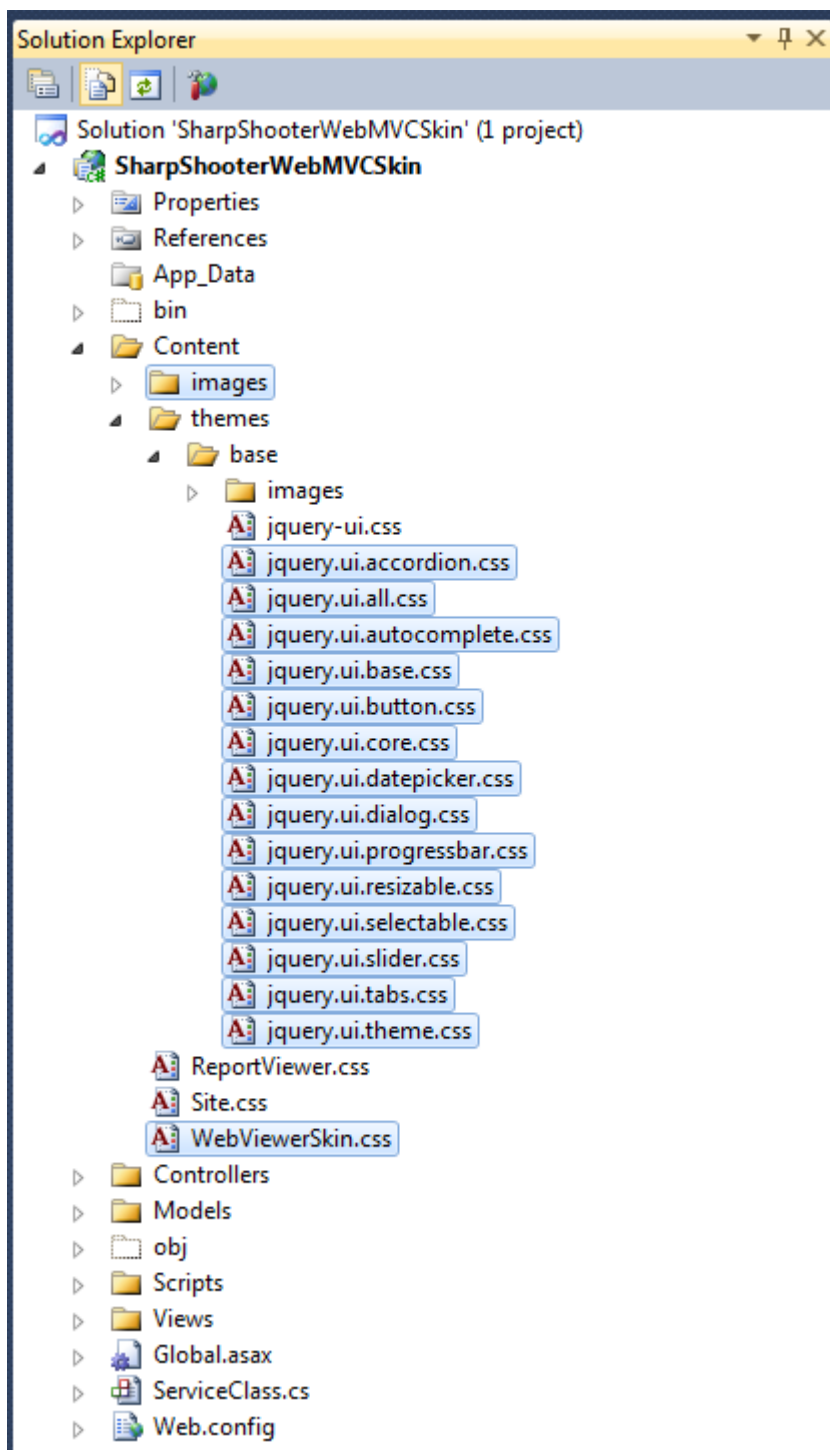
Select the “SharpShooterWebViewerSkin” project in the “Solution Explorer” and click the “Show All Files” button to preview all files in the project.



Select all the files, which are not added to the project, in the “Content” folder. Press “Include In Project” in the context menu.



After this, the files and nested files will be added to the project.

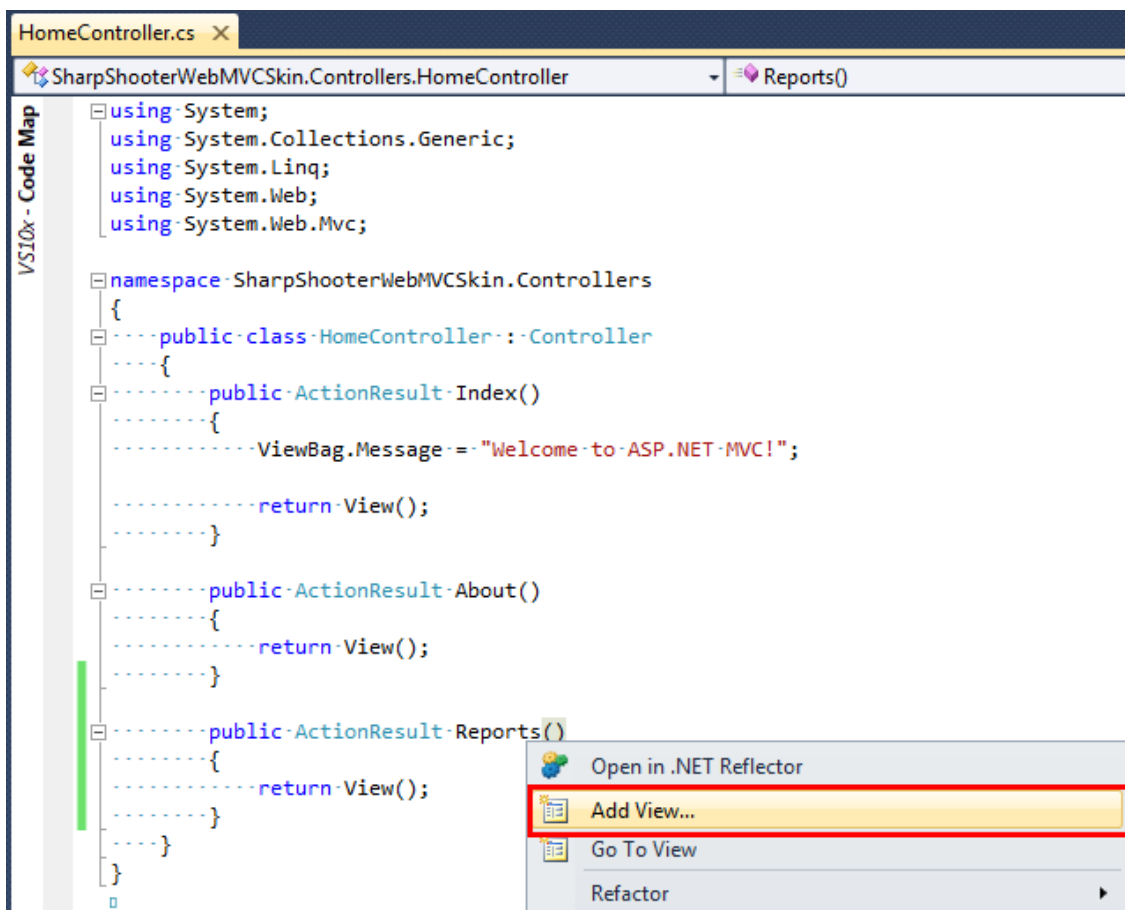


### Step 9. Adding “View” to the project

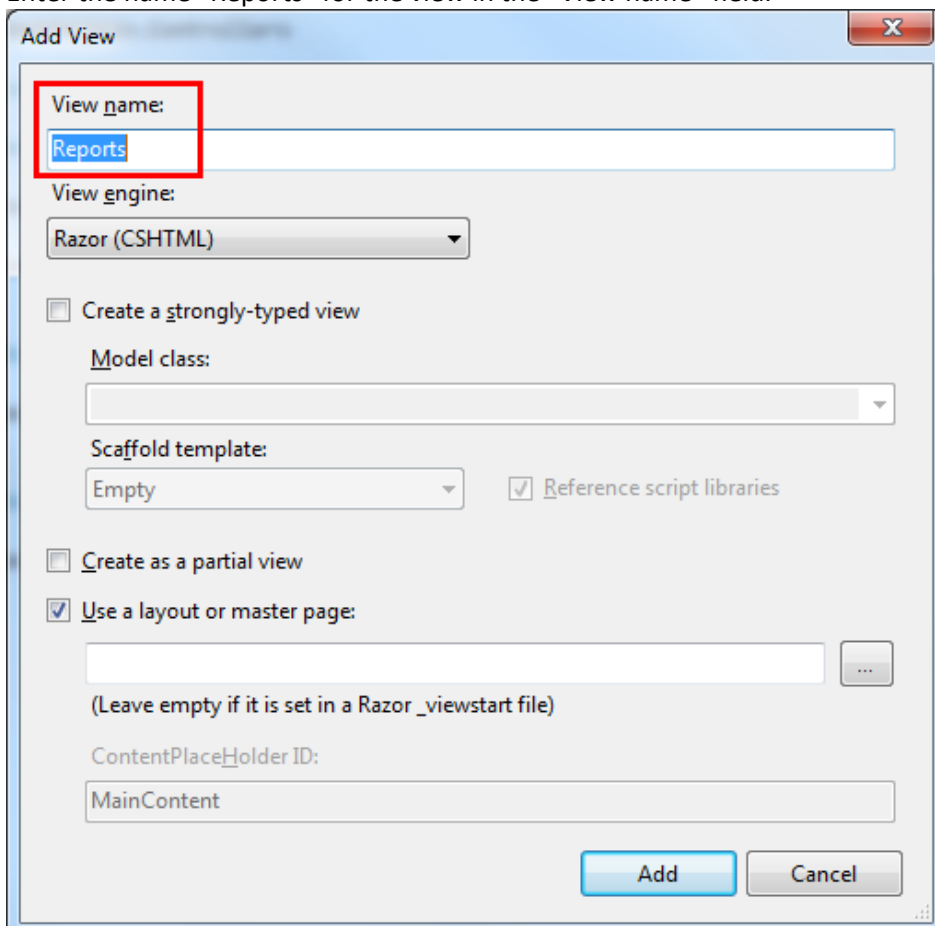
Add the action with the “Reports” name to the “HomeController” controller.

```
public ActionResult Reports()  
{  
    return View();  
}
```

Add view. Right click on the “Reports” action. Select “Add view...” in the context menu.

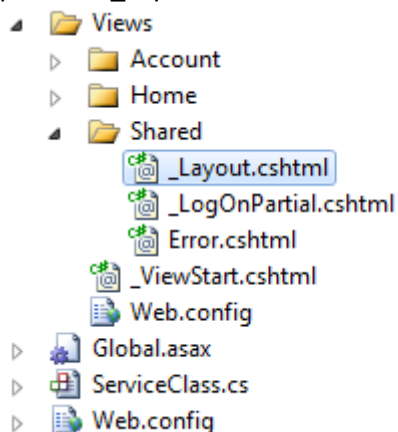


Enter the name "Reports" for the view in the "View name" field.



## Step 10. Adding scripts and styles to the view

Open the “\_Layout.cshtml” file:



Change the string:

```
<script src="@Url.Content("~/Scripts/jquery-1.4.4.min.js")" type="text/javascript"></script>
```

To:

```
<script src="@Url.Content("~/Scripts/jquery-1.5.1.js")" type="text/javascript"></script>
```

Add the code to refer to the styles:

```

<link href="@Url.Content("~/Content/themes/base/jquery.ui.all.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/ReportViewer.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/WebViewerSkin.css")" rel="stylesheet" type="text/css" />

```

Add the **link** to the “Reports.cshtml” page in the “\_Layout.cshtml” page menu.

```
<li>@Html.ActionLink("Reports", "Reports", "Home")</li>
```

The markup of the “\_Layout.cshtml” looks as follows:

```

<!DOCTYPE html>
<html>
<head>
<title>@ViewBag.Title</title>
<link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/themes/base/jquery.ui.all.css")" rel="stylesheet" type="text/css" />

```



```

<link href="@Url.Content("~/Content/ReportViewer.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/WebViewerSkin.css")" rel="stylesheet" type="text/css" />
<script src="@Url.Content("~/Scripts/jquery-1.5.1.js")" type="text/javascript"></script>
</head>
<body>
  <div class="page">
    <div id="header">
      <div id="title">
        <h1>My MVC Application</h1>
      </div>
      <div id="logindisplay">
        @Html.Partial("_LogOnPartial")
      </div>
      <div id="menucontainer">
        <ul id="menu">
          <li>@Html.ActionLink("Home", "Index", "Home")</li>
          <li>@Html.ActionLink("About", "About", "Home")</li>
          <li>@Html.ActionLink("Reports", "Reports", "Home")</li>
        </ul>
      </div>
    </div>
    <div id="main">
      @RenderBody()
      <div id="footer">
      </div>
    </div>
  </div>
</body>
</html>

```

Open the “Reports.cshtml” file by double click on the “Reports.cshtml” file in the “Solution Explorer”. Add the following code to connect the scripts to the “Reports.cshtml” file:

```

<script src="@Url.Content("~/Scripts/jquery-1.5.1.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery-ui-1.8.11.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.treeview.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/microsoftreporting.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/PerpetuumSoft.Reporting.WebViewer.Client.Model.js")"
type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/PerpetuumSoft.Reporting.WebViewer.Client.js")"
type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/WebViewer.Style.js")" type="text/javascript"></script>

```

## Step 11. Adding the Report Viewer component to the View

Add the following code snippet to the page. It defines the default view and user interface of the component.

```

<div class="ssr_reportViewerControl">
  <!--Toolbar panel, contains print and exports button-->
  <div id="ssr_toolBarPanel" class="ssr_toolBar">
    <div class="ssr_toolButtonContainer">
      <div id="ssr_printButton" class="ssr_toolButton ssr_printButtonDisable" title="Print">
      </div>
    </div>
    <div class="ssr_toolButtonContainer">
      <div id="ssr_rtfExportButton" class="ssr_toolButton ssr_rtfButtonDisable" title="Export to Rtf">
      </div>
    </div>
    <div class="ssr_toolButtonContainer">
      <div id="ssr_pdfExportButton" class="ssr_toolButton ssr_pdfButtonDisable" title="Export to Pdf">
      </div>
    </div>
    <div class="ssr_toolButtonContainer">

```



```

    <div id="ssr_excelExportButton" class="ssr_toolButton ssr_excelButtonDisable" title="Export to
Excel">
    </div>
</div>
<div class="ssr_toolButtonContainer">
    <div id="ssr_htmlExportButton" class="ssr_toolButton ssr_htmlButtonDisable" title="Export to
Html">
    </div>
</div>
<div class="ssr_toolButtonContainer">
    <div id="ssr_xpsExportButton" class="ssr_toolButton ssr_xpsButtonDisable" title="Export to Xps">
    </div>
</div>
</div>
<div class="ssr_centralDiv">
    <!--Contains thumbnails and documentMap elements-->
    <div class="ssr_navigationArea" id="ssr_navigationAreaElement">
        <div class="ssr_navigationIconPanel" id="ssr_navigationIconPanel">
            <div class="ssr_thumbnailButtonContainer" id="ssr_thumbnailButtonContainer">
                <!--shows or hides thumbnail element-->
                <div title="Thumbnails" id="ssr_thumbnailButton" class="ssr_thumbnailButton
ssr_thumbnailButtonDisable">
                </div>
            </div>
            <div class="ssr_documentMapButtonContainer" id="ssr_documentMapButtonContainer">
                <!--shows or hides document map element-->
                <div title="Document Map" id="ssr_documentMapButton" class="ssr_documentMapButton
ssr_documentMapButtonDisable">
                </div>
            </div>
        </div>
        <div id="ssr_thumbnailsPanel" class="ssr_thumbnailsPanel">
            <div id="ssr_thumbnailHeadPanel" class="ssr_thumbnailHeadPanel">
                <span class="ssr_thumbnailHeadPanelTitle">Thumbnails</span>
                <div title="Close" id="ssr_closeThumbnailPanelButton"
class="ssr_closeThumbnailPanelButton">
                </div>
            </div>
            <div id="ssr_wrapThumbnailContentPanel" class="ssr_wrapThumbnailContentPanel">
                <!--element which contains a list of thumbnail-->
                <div id="ssr_thumbnailContentPanel" class="ssr_thumbnailContentPanel">
                </div>
            </div>
        </div>
        <div id="ssr_documentMapPanel" class="ssr_documentMapPanel">
            <div id="ssr_documentMapHeadPanel" class="ssr_documentMapHeadPanel">
                <span class="ssr_documentMapHeadPanelTitle">Document Map</span>
                <div title="Close" id="ssr_closeDocumentMapPanelButton"
class="ssr_closeThumbnailPanelButton">
                </div>
            </div>
            <div id="ssr_wrapDocumentMapContentPanel" class="ssr_wrapDocumentMapContentPanel">
                <!--element which contains a document map-->
                <div id="ssr_documentMapContentPanel" class="ssr_documentMapContentPanel">
                </div>
            </div>
        </div>
    </div>
    <!--reports will be shows in this element-->
    <div id="ssr_reportContent" class="ssr_reportArea">
    </div>
</div>
<!--Statusbar panel-->
<div id="ssr_statusBarPanel" class="ssr_statusPanel">
    <div id="ssr_paginationPanel" class="ssr_paginationPanel">
        <div class="ssr_statusButtonContainer ssr_statusButtonContainerBack">

```





```

        <div id="ssr_firstPageButton" class="ssr_statusButton ssr_statusButtonBack
ssr_firstPageDisable">
            </div>
        </div>
        <div class="ssr_statusButtonContainer ssr_statusButtonContainerBack">
            <div id="ssr_previousPageButton" class="ssr_statusButton ssr_statusButtonBack
ssr_previousPageDisable">
                </div>
            </div>
            <div class="ssr_statusCurrentPageContainer">
                <!--displays the current page number-->
                <input id="ssr_currentPageNumberBox" class="ssr_currentPageNumber" title="Current page" />
                <span class="ssr_pageCountOf">of</span><span> </span><span id="ssr_pageCountBox"
                    class="ssr_pageCount">0</span>
            </div>
            <div class="ssr_statusButtonContainer ssr_statusButtonContainerForward">
                <div id="ssr_nextPageButton" class="ssr_statusButton ssr_statusButtonForward
ssr_nextPageDisable">
                    </div>
                </div>
            <div class="ssr_statusButtonContainer ssr_statusButtonContainerForward">
                <div id="ssr_lastPageButton" class="ssr_statusButton ssr_statusButtonForward
ssr_lastPageDisable">
                    </div>
                </div>
            </div>
            <!--Shows or hides error dialog-->
            <div id="ssr_notificationPanel" class="ssr_notificationPanel">
                <div id="ssr_errorButton" class="ssr_errorButton ssr_errorButtonDisable">
                    </div>
            </div>
        </div>
    </div>
    <div id="ssr_errorDialog" title="Error information" style="display: none;">
        <div id="ssr_errorMessage" class="ssr_errorMessageContent">
            </div>
        </div>
    </div>

```

Add the follow javascript code to the page. It controls the loading of the document.

```

<script type="text/javascript">
    var reportViewer = null;
    $(document).ready(function () {
        Initialize();
        InitializeReportViewer();
    });
    function InitializeReportViewer() {
        reportViewer = new PerpetuumSoft.Reporting.WebViewer.Client.ReportViewer('#ssr_reportContent');
        reportViewer.setServiceUrl("http://localhost:5555/ReportService");
        reportViewer.reportName = "CustomersReport";
        reportViewer.setDocumentMapControl("#ssr_documentMapContentPanel");
        reportViewer.setThumbnailsControl("#ssr_thumbnailContentPanel");
        reportViewer.documentInfoLoadedEvent = function (pages) {
            EnableToolbarState();
            if (pages.length > 1) {
                pageCount = pages.length;
                EnableNextLastPage();
                DisablePreviousFirstPage();
            }
            else {
                DisableNextLastPage();
                DisablePreviousFirstPage();
            }
            $("#ssr_pageCountBox").text(pages.length.toString());
            reportViewer.loadThumbnails();
        };
    };

```



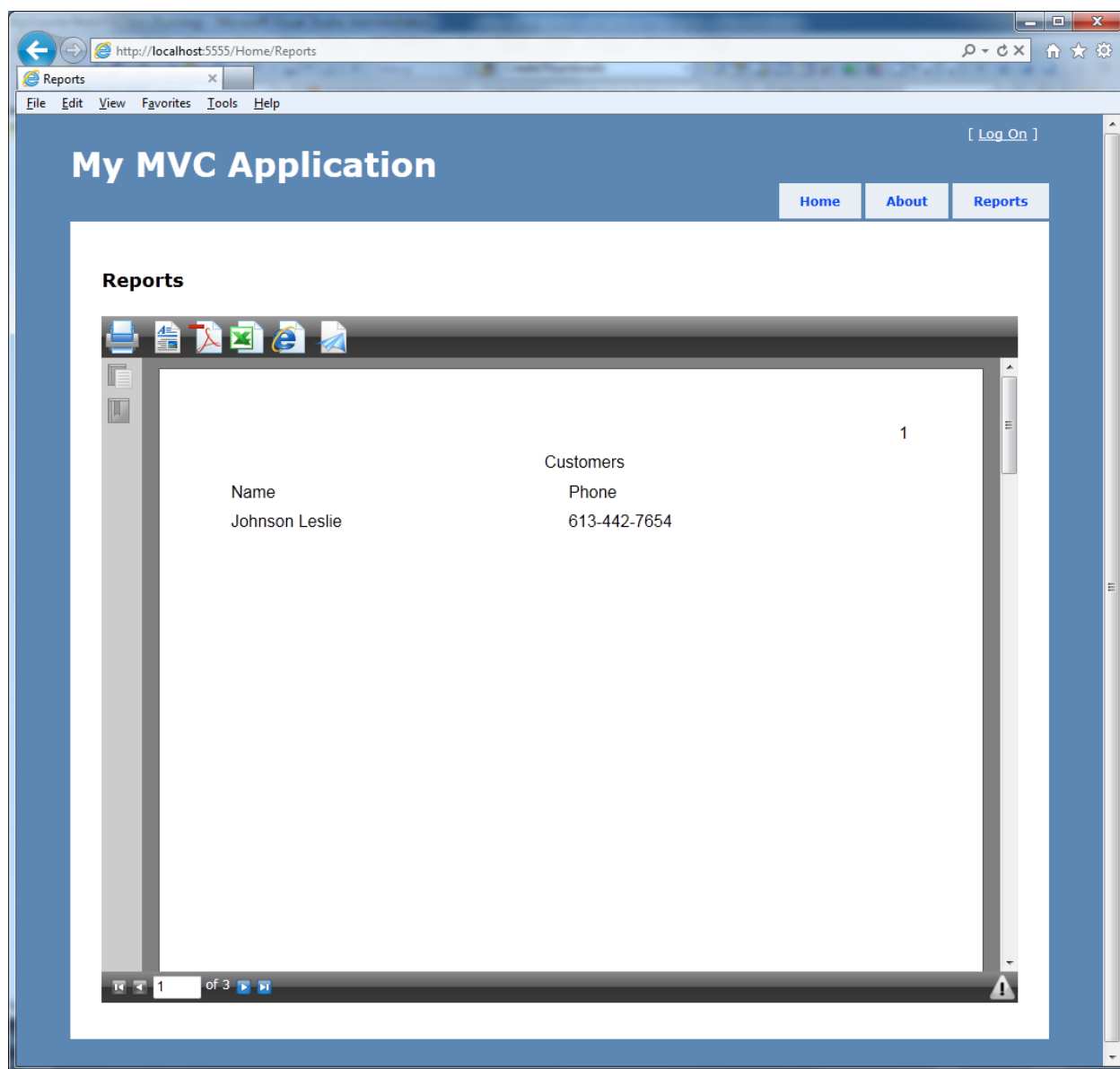
```
reportViewer.errorEvent = function (errorModel) {
    switch (errorModel.errorType) {
        case PerpetuumSoft.Reporting.WebViewer.Client.ErrorType.communicationError:
            $("#ssr_errorMessage").text(errorModel.error._error$1);
            break;
        case PerpetuumSoft.Reporting.WebViewer.Client.ErrorType.clientError:
            $("#ssr_errorMessage").text("clientError" + (errorModel.error).message);
            break;
        case PerpetuumSoft.Reporting.WebViewer.Client.ErrorType.serverError:
            $("#ssr_errorMessage").text("serverError" + (errorModel.error).message +
(errorModel.error).getInformation());
            break;
        default:
            $("#ssr_errorMessage").text(errorModel.error.message);
            break;
    }
    EnableErrorButton();
    ShowErrorDialog();
};
reportViewer.currentPageChangedEvent = function (pageNumber) {
    if (reportViewer.isFirstPage()) {
        DisablePreviousFirstPage();
    }
    else {
        EnablePreviousFirstPage();
    }
    if (reportViewer.isLastPage()) {
        DisableNextLastPage();
    }
    else {
        EnableNextLastPage();
    }
    $("#ssr_currentPageNumberBox").val(pageNumber.toString());
};

reportViewer.renderDocument();
}
</script>
```

Now your component is added to the page and you need to test it.

## Step 12. Running the application

Run the application by pressing F5 in Visual Studio. Click the “Reports” link and preview the page with the sample report.



If you have any questions regarding the integration with SharpShooter Reports.Web don't hesitate to contact us at [support@perpetuumsoft.com](mailto:support@perpetuumsoft.com).