Adding Server File Upload to the membership site

This document shows how you can add file uploading capabilities. The uploaded files will end up on the server in a folder of your choosing (~/Content/Files/).

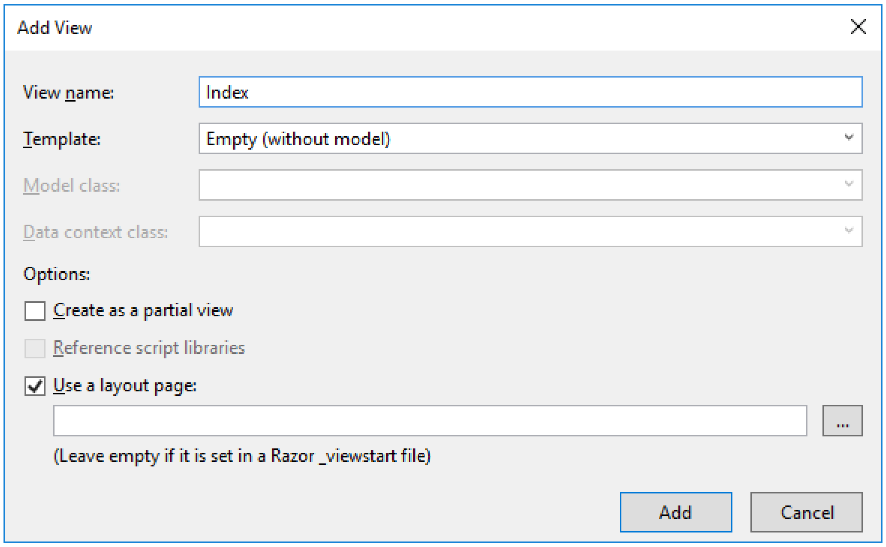
1. Open the **Content** folder in the main project.
2. Add a new folder where the uploaded files will be stored. In this example the folder will be called **Files** and have the server path **~/Content/Files/**.
3. Add a new **MVC5 Controller - Empty** controller called **File** to the **Controllers** folder in the main project.
4. Add the file list load process to the **Index** action.

   ```
   public ActionResult Index()
   {
     var path = Server.MapPath("~/Content/Files/");
     var dir = new DirectoryInfo(path);
     var files = dir.EnumerateFiles().Select(f => f.Name);
     return View(files);
   }
   ```

5. Add another **Index** action to the **FileController** class and add the **HttpPost** attribute to it. This action will be called when the **Upload** button is clicked. Add the file upload process to the action.

   ```
   [HttpPost]
   public ActionResult Index(HttpPostedFileBase file)
   {
     var path = Path.Combine(Server.MapPath("~/Content/Files/"), file.FileName);
     var data = new byte[file.ContentLength];
     file.InputStream.Read(data, 0, file.ContentLength);
     using (var sw = new FileStream(path, FileMode.Create))
     {
       sw.Write(data, 0, data.Length);
     }
     return RedirectToAction("Index");
   }
   ```

6. Right click on one of the **Index** action methods and select **Add View** in the menu to create the necessary **Index** view. The **Index** view will be added to the **Views-File** folder.



Add View

View name: Index

Template: Empty (without model)

Model class:

Data context class:

Options:

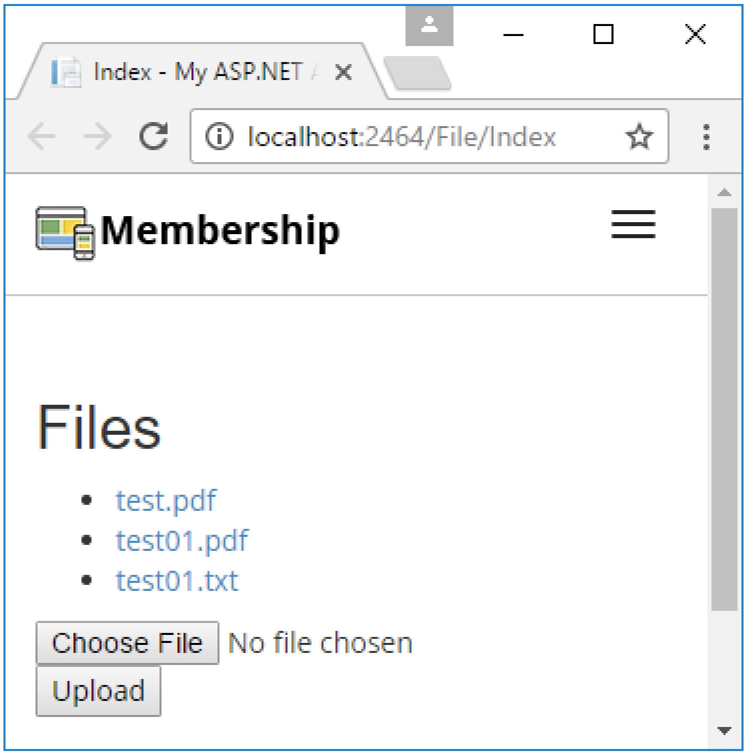
☐ Create as a partial view

☐ Reference script libraries

☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

7. Add the HTML markup to display load and upload buttons that call the desired actions. The **<ul>** element lists all files in the **~/Content/Files** folder; each **<li>** element will have a link to the file. The **<input>** element displays the **Choose file** button used to select a file from the file system. The **<button>** element calls the **HttpPost** action to upload the selected file.



```
@model IEnumerable<string>

@{
  ViewBag.Title = "Index";
}

<h2>Files</h2>

<ul>
  @foreach (var fName in Model)
  {
    var name = fName;
    var link = @Url.Content("~/Content/Files/") + name.Replace(" ", "%20");
    <li>
      <a href="@link">@name</a>
    </li>
  }
</ul>

<div>
  @using (Html.BeginForm("Index", "File", FormMethod.Post, new { enctype =
  "multipart/form-data" }))
  {
    <input type="File" name="file" id="file" value="Choose File" />
    <button type="submit">Upload</button>
  }
</div>
```

8. Run the application and navigate to the **File** Controller's **Index** action (http://localhost:XXXX/File/Index). Click the **Choose File** button and select a file from your computer's file system. Then click the **Upload** button to upload the file to the server.
9. When the view has been re-rendered you can click on the file link to display the file in the browser.