عنوان: **آموزش Backload (آپلود چندین فایل به طور همزمان با آجاکس)**
نویسنده: شقایق اشتری
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۴:۳۵
آدرس: www.dotnettips.info
گروه‌ها: MVC

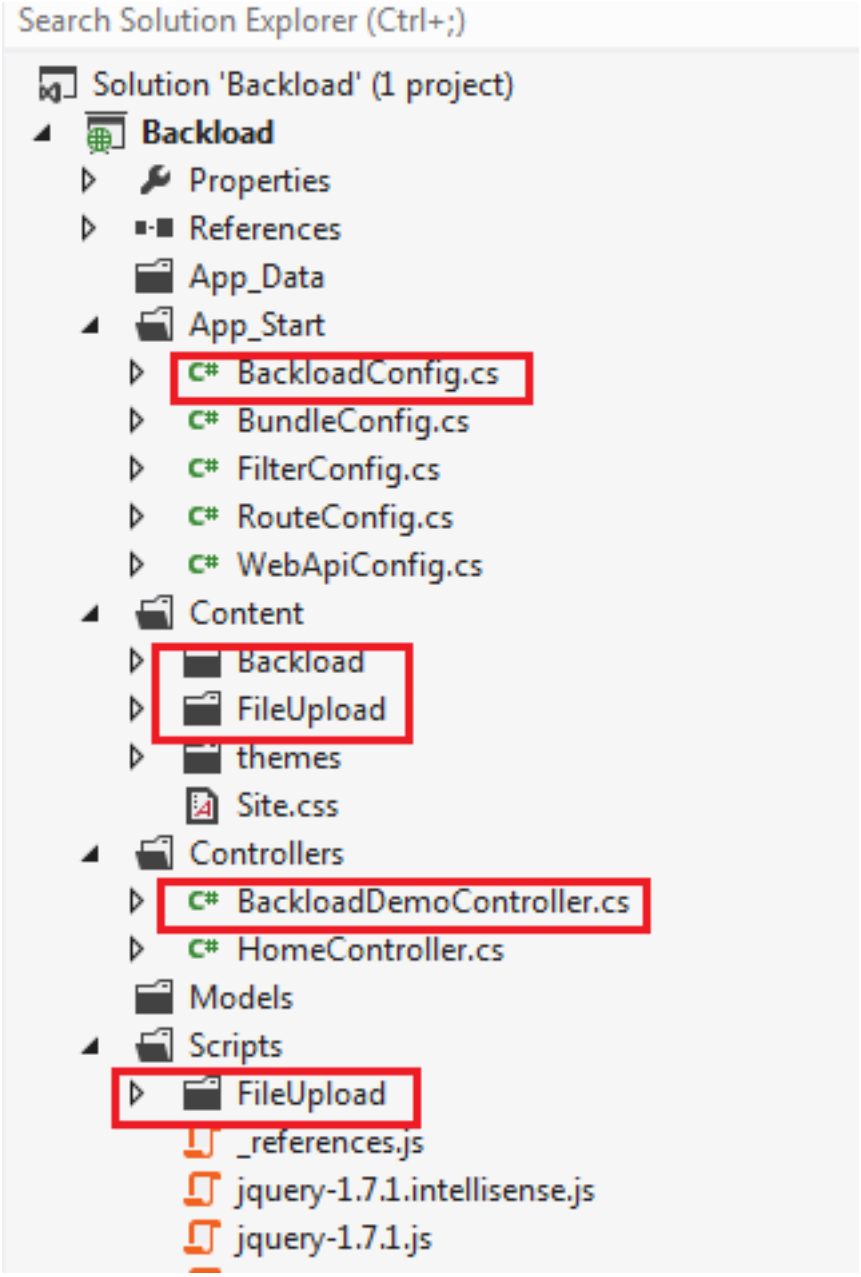
یک پروژه‌ی جدید Asp.net MVC ایجاد کنید و .Net Framework آن را 4.5 و یا بالاتر انتخاب کنید. دلیل اینکار را در ادامه‌ی آموزش به شما توضیح خواهم داد.
برای شروع کار با Backload ابتدا به قسمت Nutget می‌رویم که در مسیر زیر است :

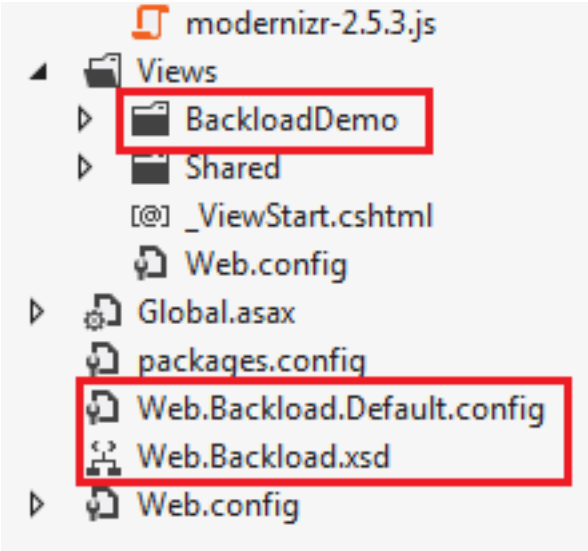
```
Tools/Library Package Manager/Manage Nutget Packages For Solution
```

در پنجره‌ی باز شده از کادر سمت چپ، بر روی قسمت Online کلیک می‌کنیم و سپس در کادر Search در گوشه‌ی بالا سمت راست، کلمه‌ی Backload را تایپ می‌کنیم در نتایج Search دو مورد زیر را Install می‌کنید :

* JQuery File Upload Plugin * Backload. A Proffesional Full Featured Asp.Net FileUpload Controller

پس از نصب دو مورد بالا ، موارد زیر را که در دو عکس پایین می‌بینید، به پروژه‌ی شما اضافه خواهند شد:


Search Solution Explorer (Ctrl+;)
Solution 'Backload' (1 project)
Backload
Properties
References
App_Data
App_Start
BackloadConfig.cs
BundleConfig.cs
FilterConfig.cs
RouteConfig.cs
WebApiConfig.cs
Content
Backload
FileUpload
themes
Site.css
Controllers
BackloadDemoController.cs
HomeController.cs
Models
Scripts
FileUpload
_references.js
jquery-1.7.1.intellisense.js
jquery-1.7.1.js


modernizr-2.5.3.js
Views
BackloadDemo
Shared
_ViewStart.cshtml
Web.config
Global.asax
packages.config
Web.Backload.Default.config
Web.Backload.xsd
Web.config

تا اینجا ما ابزار Backload را بر روی پروژه‌ی خود نصب کردیم. همانطور که می‌بینید Backload یک Demo به پروژه اضافه کرده که شامل Controller ، View و ... می‌باشد. اکنون پروژه را اجرا کنید و با FileUpload کار کنید. البته توجه داشته باشید که url صفحه‌ای که FileUpload در آن قرار دارد به این صورت است : localhost:PortNumber/BackloadDemo/Index

***نکاتی که باید بدانید:**

عکس هایی که شما آپلود می‌کنید در پوشه‌ی Files موجود در ریشه‌ی پروژه‌ی شما قرار می‌گیرند این پوشه بوسیله‌ی خود ابزار Backload ساخته می‌شود.

چنانچه بخواهید در پوشه‌ی Files پوشه‌ای ایجاد کنید، ابتدا View آن‌را باز کنید. این View در پروژه، در مسیر /Views BackloadDemo/Index قرار دارد .

در داخل تگ form یک Hidden Field با نام objectContext اضافه می‌کنید. Value ایی که شما به این Hidden Field نسبت می‌دهید، نام پوشه‌ی شما در پوشه‌ی Files خواهد بود. مانند تصویر زیر: در اینجا پوشه‌ای با نام 02-2014 در پوشه‌ی Files وقتی که فایلی را با File Upload آپلود می‌کنیم، ایجاد می‌شود.

```
orm id="fileupload" action="/Backload/UploadHandler" method="POST" enctype="multipart/form-data">
  <!-- Redirect browsers with JavaScript disabled to the origin page -->
  <noscript><input type="hidden" name="redirect" value="/"></noscript>
      <input type="hidden" name="objectContext" value="2014-02" />
  <!-- The fileupload-buttonbar contains buttons to add/delete files and start/cancel the upload --
  <div class="row fileupload-buttonbar">
```

چنانچه بخواهید در پوشه‌ای که خودتان ایجاد کردید (که در این مثال 02-2014 می‌باشد) پوشه‌های متعدد دیگری هم داشته باشید Hidden Field ای با نام uploadContext ایجاد می‌کنید؛ مانند تصویر زیر :

```
form id="fileupload" action="/Backload/UploadHandler" method="POST" enctype="multipart/form-data">
    <!-- Redirect browsers with JavaScript disabled to the origin page -->
    <noscript><input type="hidden" name="redirect" value="/"></noscript>
        <input type="hidden" name="objectContext" value="2014-02" />
      <input type="hidden" name="uploadContext" value="User200;Id20121;Images" />
    <!-- The fileupload-buttonbar contains buttons to add/delete files and start/cancel the upload -
```

اکنون اگر فایل جدیدی را آپلود کنید در مسیر

2014-02/User200/Id20121/Images

ذخیره می‌شود . یعنی بین نام هر پوشه از سمی کولن ; در Value استفاده می‌کنید.

تا اینجا ما می‌توانیم بوسیله‌ی ابزار Backload عکس‌ها را آپلود ، حذف و مسیر آپلود عکس‌ها را تغییر دهیم.

اگر توجه کرده باشید دفعات بعد که پروژه را اجرا می‌کنید عکس‌های موجود در پوشه، در گرید ویو File Upload به شما نمایش داده خواهد شد. حال اگر بخوایم عکس‌های موجود در پوشه‌ی دیگری را نمایش دهیم باید چه کار کنیم؟!
گاهی اوقات نیاز داریم که محتویات پوشه‌ای خاص را در گرید ویو File Upload مان نمایش دهیم. برای این کار شما باید کنترلر FileUploadController که در فایل ضمیمه در آموزش هست را در پوشه‌ی Controller پروژه‌ی خود کپی کنید .اگر شما این کنترلر را باز کنید مواردی مانند کلمه کلیدی Task و async را مشاهده خواهید کرد. این موارد در .Net Framework 4 شناسایی نمی‌شود. برای همین در ابتدای آموزش تاکید کردم که .Net Framework 4.5 و یا بالاتر را برای پروژه‌ی خود انتخاب کنید .
در تابع Handler_GetFilesRequestStarted در این کنترلر باید مشخص کنید که فایل‌های موجود در کدام مسیر را برای شما نمایش دهد؛ آن هم با استفاده از دستور زیر :

```
// Begin of core GET execution method, values of the GET request can be edited, or GET request can be aborted
void handler_GetFilesRequestStarted(object sender, Eventing.Args.GetFilesRequestEventArgs e)
{
    // Demo 4: Allow only jpg files
    // e.Param.BackloadValues.FilesFilter = "*.jpg";
    e.Param.SearchPath = Server.MapPath("~/Files/2014-02/User200/Id20121/Images");
    e.Context.PipelineControl.Message.MessageText += string.Format(_logpattern, "log-get", "GetFilesRequestStar
}
```

اکنون قبل از آنکه پروژه را اجرا کنید فایل Backload.Demo.js که در مسیر Scripts/Fileupload هست را باز کنید و url موجود در آنرا مانند عکس زیر تغییییر دهید :

backload.demo.js

```
/*
 * jQuery File Upload Plugin JS Example 8.0.1
 * https://github.com/blueimp/jQuery-File-Upload
 *
 * Copyright 2010, Sebastian Tschan
 * https://blueimp.net
 *
 * Licensed under the MIT license:
 * http://www.opensource.org/licenses/MIT
 */

/*jslint nomen: true, regexp: true */
/*global $, window, navigator */

$(function () {
    'use strict';

  //  var url = '/Backload/UploadHandler';
    var url = '/FileUpload/FileHandler';
    // Initialize the jQuery File Upload widget:
    $('#fileupload').fileupload({
        // Uncomment the following to send cross-domain cookies:
        //xhrFields: {withCredentials: true},
        url: url
    });

    // Enable iframe cross-domain access via redirect option:
    $('#fileupload').fileupload(
        'option',
        'redirect',
        window.location.href.replace(
            /\/[^\/]*$/,
```

حالا پروژه را اجرا کنید. خواهید دید تمامی فایل‌های موجود در مسیری را که شما مشخص کرده‌اید، برایتان نمایش خواهد داد.

چنانچه بخواهید تعداد مثلا 5 فایل برای شما در گرید ویو مربوط به FileUpload نمایش داده شود، به تابع handler_GetFilesRequestFinished می‌روید. متغیر limit مشخص می‌کند که 5 فایل نمایش داده شود. می‌توانید این عدد را به دلخواه خود تغییر دهید.

نکته‌ی بسیار مهم دیگری که باید به آن توجه شود مربوط به Hidden Field نام پوشه‌ها می‌باشد. بار دیگر پروژه را اجرا کنید. با استفاده از ابزاری مثل FireBug کدهای html صفحه‌ی خود را ببینید. Hidden Field ایی با نام objectContext را پیدا کنید و Value آنرا به test تغییر دهید. فایلی را آپلود کنید حالا به پوشه‌ی Files موجود در ریشه‌ی پروژه بروید. می‌بینید که پوشه‌ای با نام testایجاد شده و فایلی هم که آپلود کردید در آن قرار دارد که این یک اشکال است. برای اینکه جلوی این گونه کارها را بگیریم به تابع handler_StoreFileRequestStartedAsync می‌رویم و کد زیر را می‌نویسیم :

```
async Task handler_StoreFileRequestStartedAsync(object sender, Eventing.Args.StoreFileRequestEventArgs e)
{
    if (e.Param.FileStatusItem.StorageInfo.FilePath != Server.MapPath("YourPath"))
    {
        e.Context.PipelineControl.ExecutePipeline = false;
    }
        e.Context.PipelineControl.Message.MessageText += string.Format(_logpattern, "log-post", "StoreFileRe
    }
```

دستور e.Context.PipelineControl.ExecutePipeline = false; باعث میشود که اجرای تابع متوقف شود.
فایل ضمیمه :
FileUploadController-462d551688cf48c68cb55343ac5464f3.zip

برای مشاهده مثال‌های دیگری درباره‌ی Backload به این لینک بروید.

موفق باشید.

این نوشتار اولین آموزش من در این سایت می‌باشد و جا دارد از دوست خوبم "محبوبه قربانی" که باعث شد من با MVC آشنا شوم تشکر کنم.

نظرات خوانندگان

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۶:۸

با تشکر از مطلب مفید شما
این کتابخانه داره از کتابخانه ی jQuery File Upload استفاده می‌کنه. به نظرم کسی نحوه‌ی استفاده از این کتاب خانه را با توجه به sampleهای خود سایتش یاد بگیره خیلی بهتره. نسخه‌ی angularjs هم داره.

نویسنده: شقایق اشتری
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۶:۵۰

ممنونم .
نوشته‌ی من در حد یک آشنایی و معرفی با این ابزار بود .
و اینکه بهتره مقاله‌ی بنده را دقیق‌تر بخوانید چون آخر مقاله لینک مثال‌های کار با backload را گذاشتم.
مورد بعدی اینکه متاسفانه دستوراتی مثل e.param.Searchpath را در مثال‌های آماده‌ی خود سایت هم به آن اشاره ای نکرده بود و من خودم فهمیدم که این دستور را باید در کدام تابع بنویسم تا جواب دهد .
نکته‌ی بعدی Hidden Field که به عنوان نام پوشه استفاده می‌شود اگر یک کاربر سایت ما مقدار value این hidden field را تغییر دهد اسم پوشه به کل تغییر می‌کند که در مثال‌های آماده‌ی خود سایت هم به این موضوع اشاره ای نکرده بود این راه حل هم خودم فکر کردم و به نتیجه ای رسیدم که در بالا توضیح دادم .

نویسنده: شقایق اشتری
تاریخ: ۱۳۹۳/۰۱/۲۴ ۱۳:۴۹

نکته ای بسیار مهم در رابطه با کار با Backload:

حذف و اضافه کردن تصاویر با Backload روی لوکال بدون هیچ مشکلی انجام می‌شود اما بعد از اینکه شما پروژه را آپلود کردید خواهید دید که تصاویر اضافه می‌شوند اما حذف نمی‌شوند.
برای حل این مشکل کد زیر را در تگ <system.webServer> در فایل web.config قرار دهید :

```
<modules runAllManagedModulesForAllRequests="true">
      <remove name="WebDAVModule"/>
  </modules>
```

موفق باشید .

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۳/۲۹ ۱۸:۳۵

از مطلب خوبتون ممنونم.
یه مشکلی که من بهش بر خوردم این بود که این ابزار در آپلود فایل‌های همنام ما را دوچار مشکل میکنه ، اگر بشود با کمی دستکاری هنگام آپلود ، در صورت همنام بودن یک GUID به نام آن بچسبانیم این مشکل برطرف میشد.

نویسنده: شقایق اشتری
تاریخ: ۱۳۹۴/۰۳/۳۰ ۹:۱۷

چنانچه بخواهید هنگام آپلود فایل‌ها نامشان ر ا به نام دلخواه تغییر دهید در تابع handler_StoreFileRequestStartedAsync کد زیر را بکار می‌برید

```
string Filen = e.Param.FileStatusItem.FileName;
int find = Filen.LastIndexOf(".");
e.Param.FileStatusItem.FileName = DateTime.Now.Ticks + "-" + Filen.Substring(0,find-1)+
Filen.Substring(find, Filen.Length - find);
e.Param.FileStatusItem.UpdateStatus(true);
```