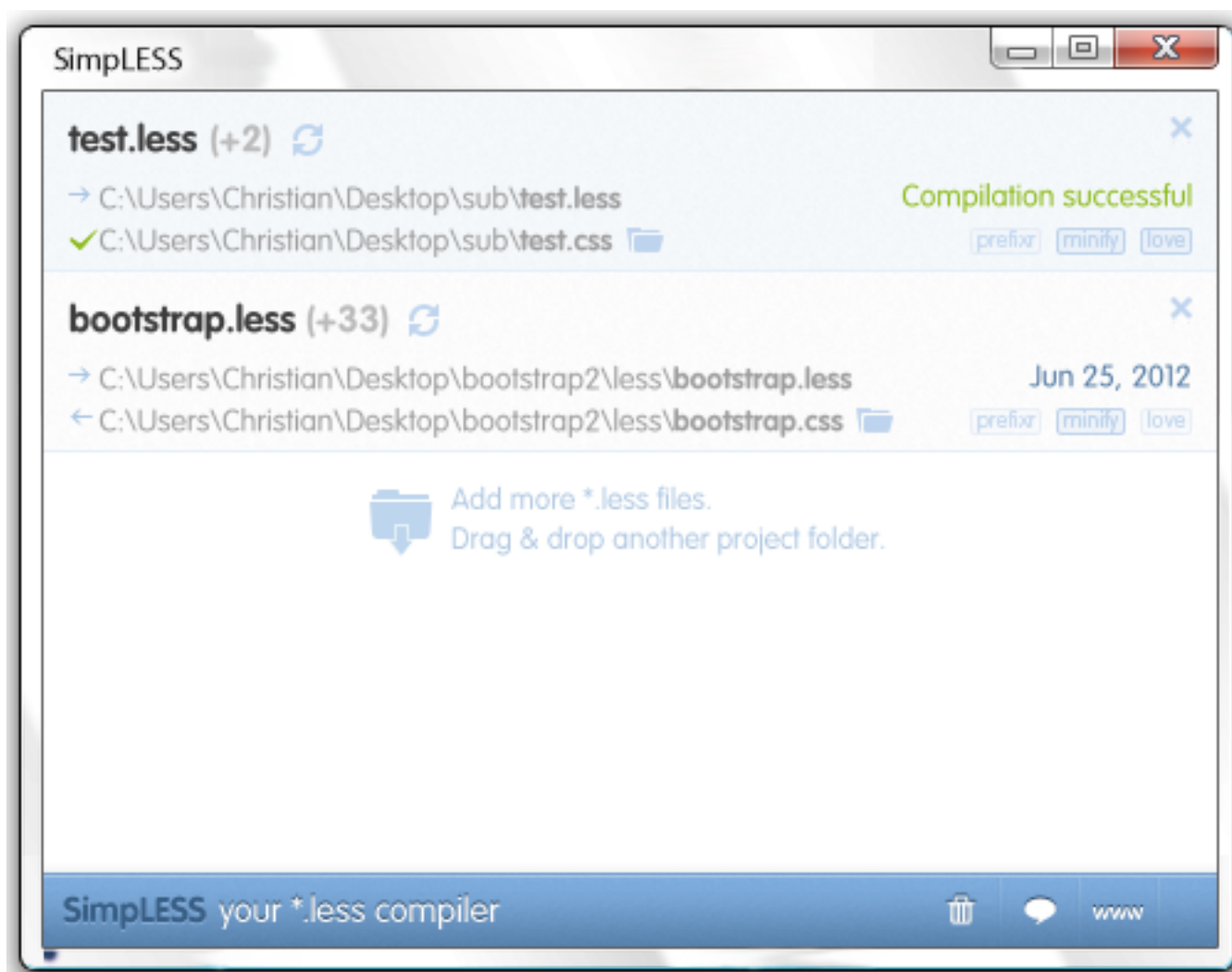


در این مطلب، نحوه‌ی استفاده از افزونه‌ی Web Essentials جهت پردازش فایل‌های LESS را بررسی می‌کنیم. پیش‌تر مطالبی را در رابطه با [CSS pre-processor](#) ها مطالعه کرده‌اید، [LESS](#) نیز یک CSS pre-processor است، یا در واقع بهتر است بگوئیم یک زبان جهت پویا کردن CSS می‌باشد که در سال 2009 توسط Alexis Sellier به صورت سورس باز [ایجاد شد](#). یکی از خصوصیات جالب LESS نسبت به دیگر CSS preprocessors، قابلیت کامپایل فایل‌های CSS به صورت real-time از طریق مرورگر توسط LESS.js می‌باشد.

روش‌های استفاده از LESS در دات نت 1- استفاده از SimpleLESS تمام تغییرات فایل‌های CSS را مشاهده می‌کند و در نهایت به صورت خودکار آنها را به CSS کامپایل و همچنین minify می‌کند. این روش مستقل از دات نت است و برای هر فایل LESSی جوابگو خواهد بود.



2- استفاده از .less{}

Dotless یک پیاده سازی از کتابخانه جاوا اسکریپتی LESS برای دات نت می‌باشد. پکیج نیوگت Dotless را نیز می‌توانید از [اینجا](#) دریافت کنید. بعد از اضافه شدن فایل‌های آن، یک ارجاع به dotless.core به پروژه تان اضافه خواهد شد. همچنین در فایل Web.Config در قسمت HttpHandler خط زیر اضافه خواهد شد:

```
<add type="dotless.Core.LessCssHttpHandler,dotless.Core" validate="false" path="*.LESS" verb="*" />
```

خط فوق یعنی به محض مواجه شدن با فایل LESS، پردازشگر فایل‌های LESS وارد عمل می‌شود. همچنین خط زیر نیز جهت پیکربندی به قسمت configSections در فایل Web.Config اضافه می‌شود:

```
<section name="dotless"  
type="dotless.Core.configuration.DotlessConfigurationSectionHandler,dotless.Core" />
```

همچنین اگر مایل بودید می‌توانید تنظیمات مربوط به فشرده سازی و caching را نیز فعال کنید:

```
<dotless minifyCss="false" cache="true" />
```

3- استفاده از افزونه‌ی Web Essentials

[Web Essentials](#) برای کامپایل فایل‌های LESS از کامپایلر node استفاده می‌کند. کار با این افزونه خیلی ساده است. کافی است پسوند فایل CSS موجود در پروژه تان را درون ویژوال استودیو، به .less تغییر دهید. با دوبار کلیک بر روی فایل، ویرایشگر فایل‌های LESS برای شما نمایش داده می‌شود، همزمان نیز فایل یک فایل CSS و یک نسخه از فایل CSS را به صورت فشرده، برایتان تولید می‌کند. خب، هر بار که فایل LESS را تغییر دهید، Web Essentials به صورت خودکار فایل‌های .css و .min.css را برایتان روز رسانی می‌کند.



خوب با کلیک بر روی فایل less، ویرایشگر فایل‌های less نمایش داده می‌شود که با تغییر فایل css می‌توانید پیش نمایش آنرا در سمت راست مشاهده کنید:



تعریف متغیر

با استفاده از syntax زیر می‌توانید متغیرهای خود را تعریف کنید:

```
@variable-name: variableValue;
```

یکی از قابلیت‌های جالب در حین مقداردهی متغیرها به خصوص زمانی که مقدار یک کد رنگی باشد، نمایش کادر انتخاب رنگ است، این کادر بلافاصله بعد از نوشتن علامت # در ابتدای مقدار متغیر نمایش داده می‌شود:



به طور مثال با تعریف متغیر فوق هر جایی می‌توانیم برای تعیین رنگ از آن استفاده کنیم:

```
@primary-color: #ff6a00;
body
{
  background-color: @primary-color;
}
```

استفاده از توابع

LESS شامل تعداد زیادی [توابع](#) از پیش نوشته شده است که می‌توانید به راحتی از آنها استفاده کنید، توابعی از جمله کار با رنگ ها، اعمال ریاضی و غیره. استفاده از آنها خیلی ساده است. به طور مثال در کد زیر از تابع percentage جهت تبدیل 0.5 به 50% استفاده کرده ایم:

```
.myClass
{
  width: percentage(0.5);
}
```

استخراج یک فایل

یکی دیگر از قابلیت‌های Web Essentials استخراج (Extract) یک فایل می‌باشد به طور مثال فایل LESS شما شامل متغیرهای زیر است:

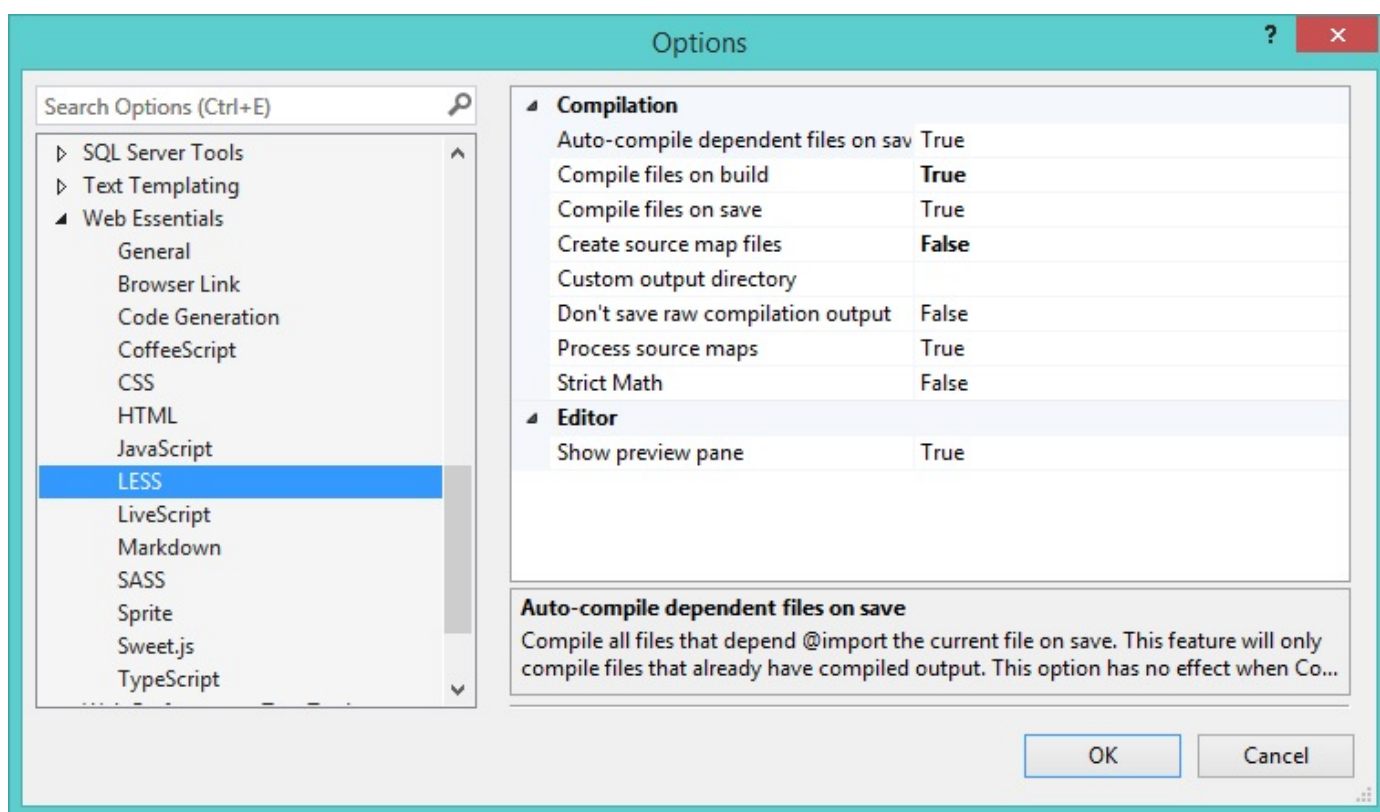
```
@primary-color: #7BA857;
@primary-color-light: #B6DE8F;
@primary-color-lighter: #D3EFC3;
@primary-color-lightest: #EFFAE6;
@secondary-color: #AE855C;
@text-color-light: #666666;
@text-color-dark: #0444;
```

به راحتی می‌توانید تعاریف فوق را درون یک فایل LESS دیگر با نام colors.less قرار دهید:



تغییر تنظیمات پیش فرض Web Essentials

افزونه Web Essentials دارای یک قسمت جهت تغییر تنظیمات پیش فرض برای کار با LESS می‌باشد که با مراجعه به منوی Tools در ویژوال استودیو و سپس Options می‌توانید آنها را تغییر دهید:



Auto-compile dependent files on save: توسط این گزینه می‌توانیم تعیین کنیم که فایل‌های که import کرده ایم تنها در صورتی که تغییر کرده و ذخیره شده باشند، در فایل CSS جاری کامپایل شوند.

Compile files on build: توسط این گزینه می‌توانیم تعیین کنیم که فایل‌های less در زمان Build پروژه کامپایل شوند.

Compile files on save: توسط این گزینه می‌توانیم تعیین کنیم که فایل‌های less در زمان ذخیره کردن پروژه کامپایل شوند.

Create source map files: اگر این گزینه True باشد [فایل map](#) نیز تولید خواهد شد.

Custom output directory: اگر می‌خواهید خروجی در پوشه‌ی موردنظر شما نمایش داده شود می‌توانید آدرس آن را تعیین کنید.

Don't save raw compilation output: با فعال بودن این گزینه فایل CSS عادی ایجاد نخواهد شد.

Process source maps: توسط این گزینه می‌توانید قابلیت‌های ویرایشگر فایل‌های source map را فعال یا غیرفعال کنید.

Strict Math: با فعال بودن این گزینه LESS تمام اعمال ریاضی درون فایل CSS را پردازش خواهد کرد.

Show preview pane: از این گزینه نیز جهت نمایش یا عدم نمایش preview window استفاده می‌شود.

نظرات خوانندگان

نویسنده: محمدرضا برنتی
تاریخ: ۴:۳۷ ۱۳۹۳/۰۴/۱۲

در قسمت روش‌های استفاده از less جای کامپایل با دستور زیر در CMD :

```
lessc style.less > style.css
```

و کامپایل توسط جاوا اسکریپت که به نظر من بهترین راه برای مشاهده تغییرات جزئی در فایل less پروژه قبل از کامپایل نهایی است خالی مانده .
برای کامپایل با جاوا اسکریپت :

```
<link rel="stylesheet/less" type="text/css" href="content/style.less" />  
<script src="Script/less.min.js"></script>
```

در خط اول مقدار خاصیت href برابر آدرس فایل less و src خط دوم شامل آدرس فایل کامپایلر جاوا اسکریپت بهینه شده است .
لینک دانلود کامپایلر جاوا اسکریپت: less.min.js

نویسنده: سیروان عقیفی
تاریخ: ۹:۰۲ ۱۳۹۳/۰۴/۱۲

هنگام کار با LESS در Visual Studio توسط افزونه‌ی Web Essentials نیازی به این کار نیست، شما کافی است فایل کامپایل شده CSSتان را داخل Layoutتان قرار دهید بعد از اینکار در هر بار تغییر فایل less و کامپایل مجدد آن(چه در حالت build و چه در حالت save) فایل اضافه شده به Layout به صورت خودکار به روز رسانی می‌گردد.

نویسنده: محمدرضا برنتی
تاریخ: ۱۲:۳۹ ۱۳۹۳/۰۴/۱۲

این افزونه با هیچ کدام از روش‌های نصبی که تو سایت Web Essentials پیشنهاد شده بود روی vs 2013 من نصب نشد و با پیغام زیر متوقف شد .

This extension is not installable on any currently installed products.

نویسنده: وحید نصیری
تاریخ: ۱۲:۵۸ ۱۳۹۳/۰۴/۱۲

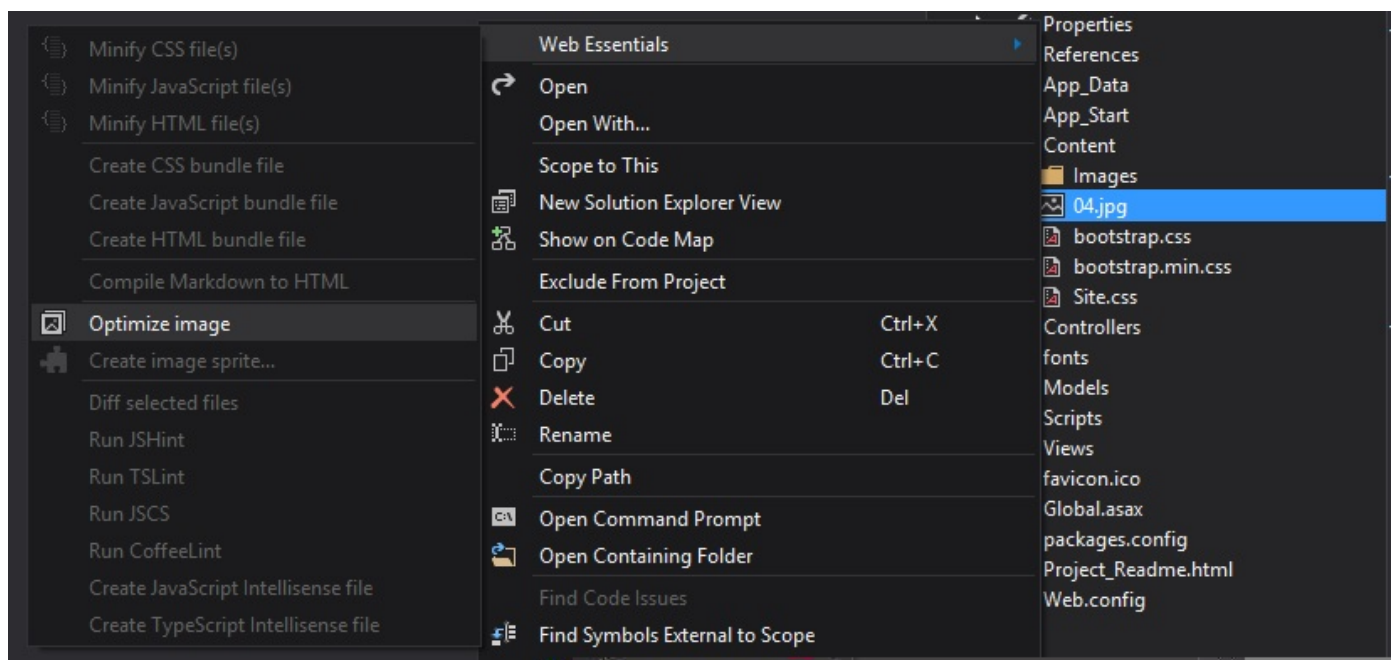
[اینجا](#) بحث شده. برای نصب آخرین نگارش آن، ابتدا [به روز رسانی دوم VS 2013](#) را نصب کنید. سپس از طریق منوی نصب افزونه‌های داخل خود VS.NET و نه دریافت مستقیم فایل vsix آن اقدام به نصب کنید.

در این مطلب نکات کار با تصاویر را توسط افزونه‌ی Web Essentials بررسی می‌کنیم. این افزونه قابلیت‌های زیر را در کار با تصاویر در اختیار شما قرار می‌دهد: **بهینه‌سازی تصاویر**

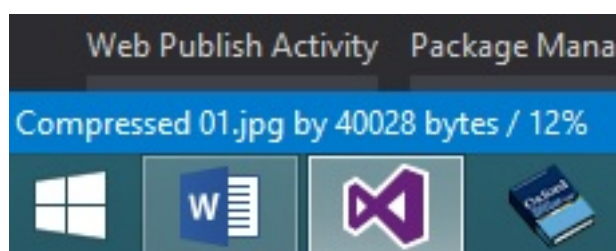
یکی از موارد مهمی که باید مورد توجه قرار بگیرد، استفاده از تصاویر کم حجم در وب‌سایت می‌باشد. روش‌های مختلفی جهت بهینه‌سازی تصاویر مورد استفاده در سایت وجود دارند، به طور مثال جهت بهینه‌سازی تصاویر PNG می‌توانید از ابزار PNGGauntlet استفاده کنید. همچنین [اینجا](#) نیز یک ابزار آنلاین موجود می‌باشد. افزونه‌ی Web Essentials این قابلیت را به آسانی در اختیار شما قرار می‌دهد؛ اینکار را می‌توانید توسط این افزونه به روش‌های زیر انجام دهید:

کلیک راست بر روی تصویر

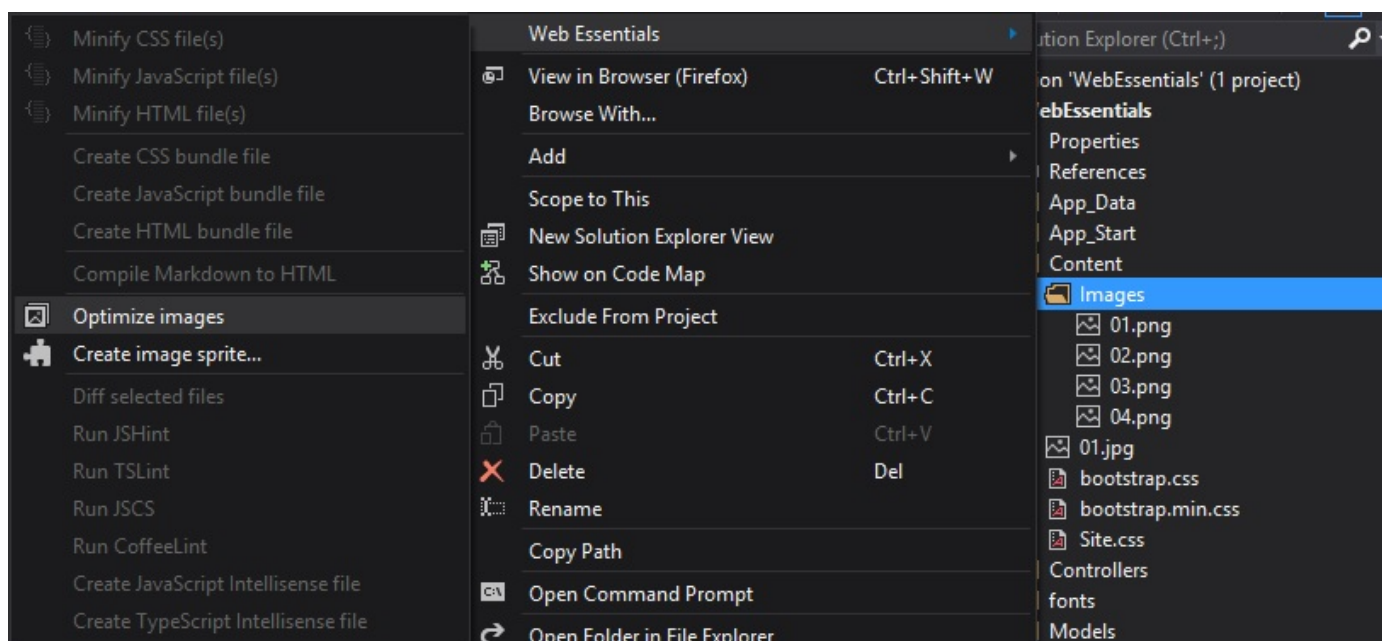
برای اینکار بر روی فایل‌ای که می‌خواهید optimize کنید، کلیک راست کرده و از منوی ظاهر شده گزینه Web Essentials و سپس Optimize Image را انتخاب کنید:



در قسمت status bar نیز می‌توانید نتیجه را مشاهده کنید:

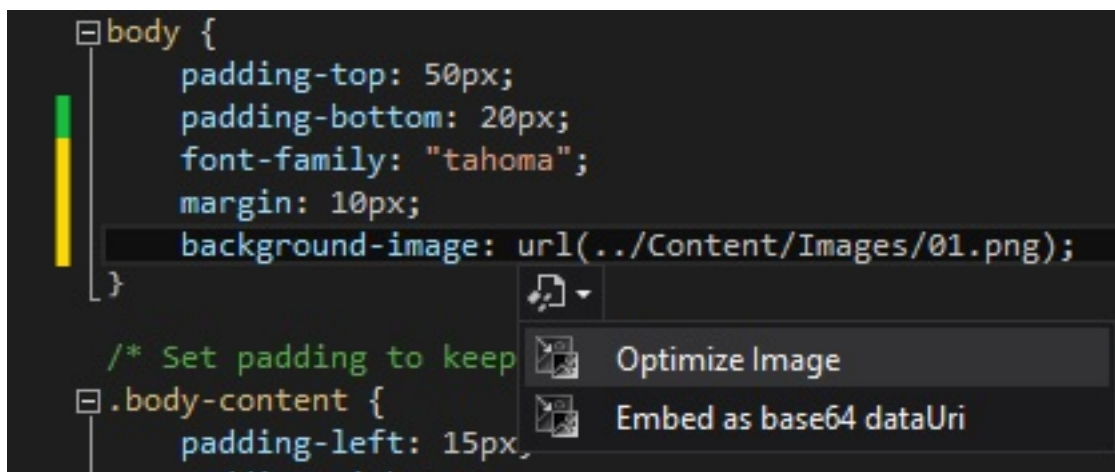


روال قبلی را می‌توانید برای چندین فایل انتخاب شده و یا یک پوشه تکرار کنید:



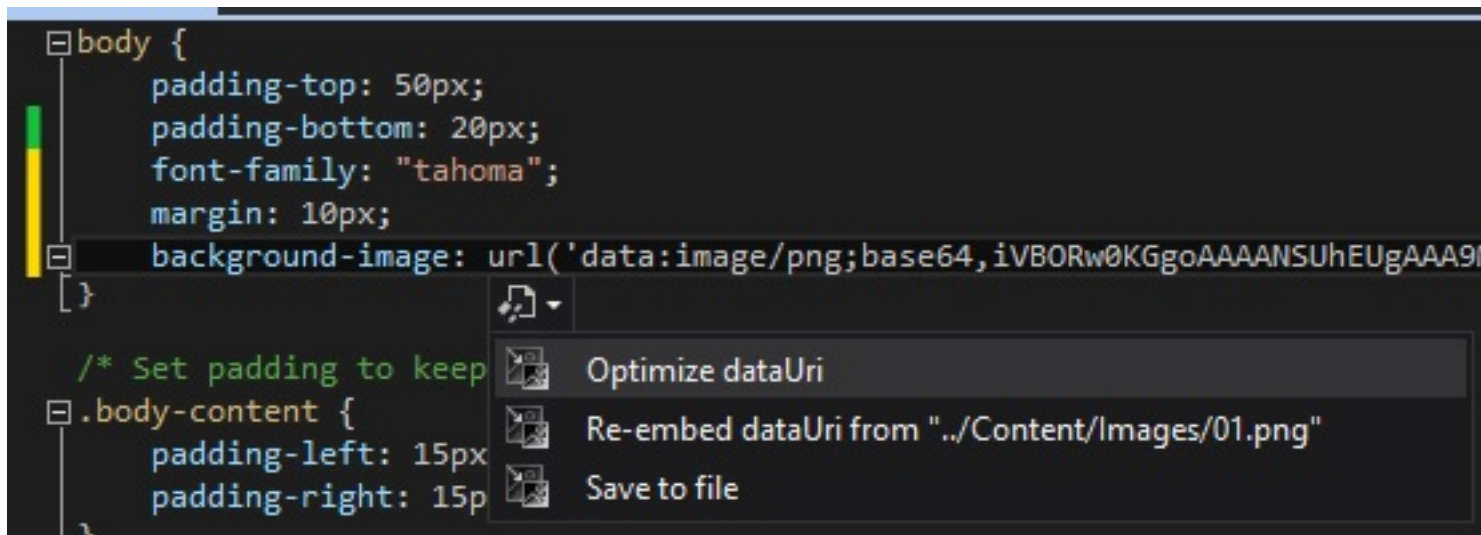
بهینه‌سازی تصاویر موجود در فایل‌های CSS

همچنین امکان بهینه‌سازی تصاویر داخل فایل‌های CSS نیز توسط این افزونه امکان پذیر است:



بهینه سازی تصاویر Base64 Encode

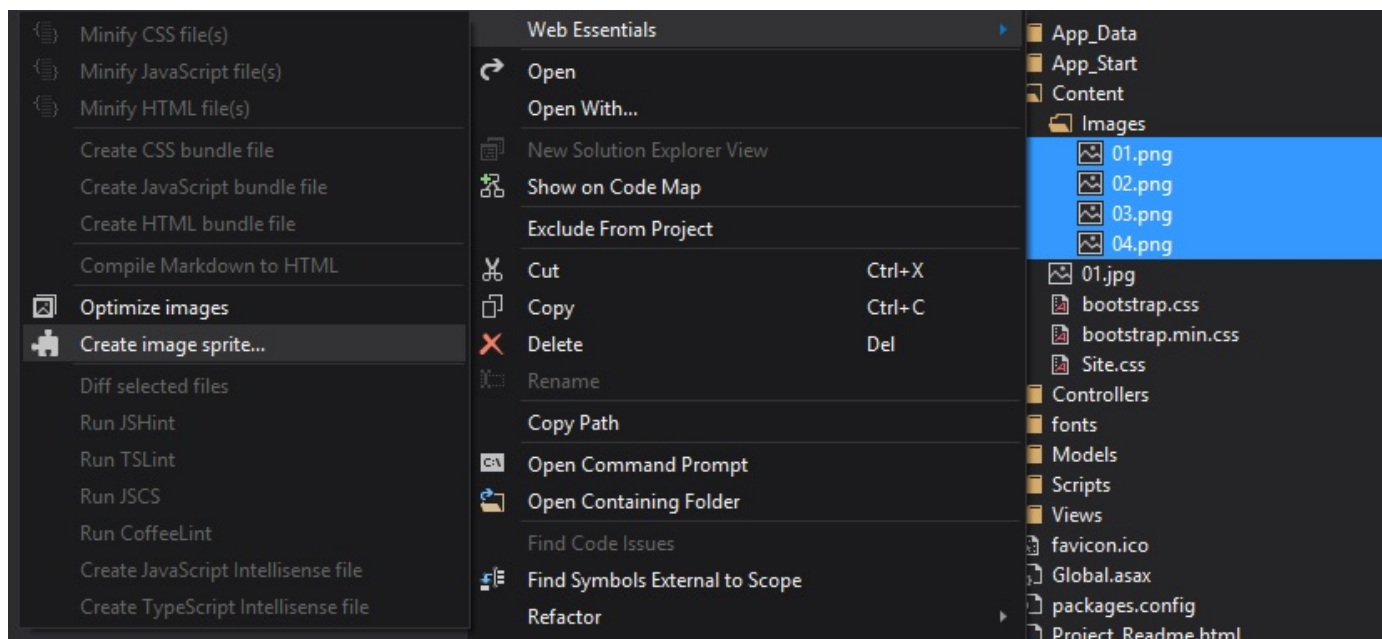
توسط این افزونه می‌توانیم تصاویر [Data Uri](#) را نیز بهینه سازی کنیم:



همانطور که در تصویر فوق مشاهده می‌کنید می‌توانیم تصاویری که به صورت Data Uri درون کد پیوست شده اند را با کلیک بر روی Save to file به صورت یک فایل ذخیره کنیم.

ایجاد تصاویر Sprite

یکی دیگر از قابلیت‌های افزونه Web Essentials امکان تهیه تصاویر به صورت [Sprite](#) می‌باشد. برای اینکار کافی است به این صورت عمل کنید:

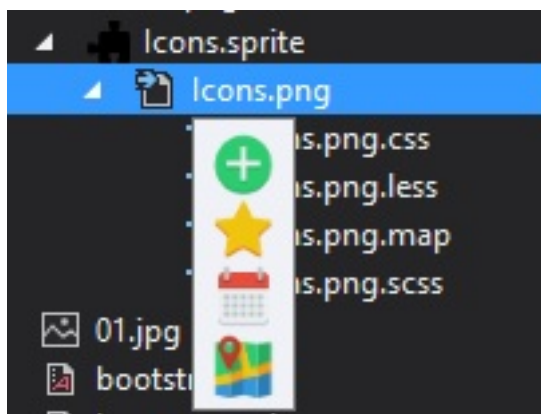


بعد از کلیک بر روی Create image sprite باید یک نام برای آن تعیین کنید و سپس بر روی کلید Save کلیک کنید. با اینکار یک فایل از نوع XML با پسوند sprite برای شما ساخته خواهد شد:

```
<?xml version="1.0" encoding="utf-8"?>
<sprite xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://vswebessentials.com/schemas/v1/sprite.xsd">
  <settings>
    <!--Determines if the sprite image should be automatically optimized after creation/update.-->
    <optimize>true</optimize>
    <!--Determines the orientation of images to form this sprite. The value must be vertical or
```

```
horizontal.-->
<orientation>vertical</orientation>
<!--File extension of sprite image.-->
<outputType>png</outputType>
<!--Determin whether to generate/re-generate this sprite on building the solution.-->
<runOnBuild>>false</runOnBuild>
<!--Use full path to generate unique class or mixin name in CSS, LESS and SASS files. Consider
disabling this if you want class names to be filename only.-->
<fullPathForIdentifierName>>true</fullPathForIdentifierName>
<!--Use absolute path in the generated CSS-like files. By default, the URLs are relative to sprite
image file (and the location of CSS, LESS and SCSS).-->
<useAbsolutePath>>false</useAbsolutePath>
<!--Specifies a custom subfolder to save CSS files to. By default, compiled output will be placed
in the same folder and nested under the original file.-->
<outputDirectoryForCss />
<!--Specifies a custom subfolder to save LESS files to. By default, compiled output will be placed
in the same folder and nested under the original file.-->
<outputDirectoryForLess />
<!--Specifies a custom subfolder to save SCSS files to. By default, compiled output will be placed
in the same folder and nested under the original file.-->
<outputDirectoryForScss />
</settings>
<!--The order of the <file> elements determines the order of the images in the sprite.-->
<files>
<file>/Content/Images/01.png</file>
<file>/Content/Images/02.png</file>
<file>/Content/Images/03.png</file>
<file>/Content/Images/04.png</file>
</files>
</sprite>
```

یکی از زیر مجموعه‌های این فایل، تصویر نهایی می‌باشد، همچنین فایل‌های map، less، css و scss آن نیز تولید می‌شود:



به عنوان مثال فایل CSS تصویر فوق به صورت زیر می‌باشد:

```
/*
This is an example of how to use the image sprite in your own CSS files
*/
.Content-Images-01 {
/* You may have to set 'display: block' */
width: 32px;
height: 32px;
background: url('icons.png') 0 0;
}
.Content-Images-02 {
/* You may have to set 'display: block' */
width: 32px;
height: 32px;
background: url('icons.png') 0 -32px;
}
.Content-Images-03 {
/* You may have to set 'display: block' */
width: 32px;
height: 32px;
background: url('icons.png') 0 -64px;
```

```
}  
.Content-Images-04 {  
/* You may have to set 'display: block' */  
width: 32px;  
height: 32px;  
background: url('icons.png') 0 -96px;  
}
```

هر کدام از کلاس‌های فوق به یک تصویر در فایل مربوطه توسط image position اشاره می‌کند. شما می‌توانید با انتساب هر کدام از کلاس‌های فوق به یک المنت از آن تصویر استفاده نمایید:

```
<div class="Content-Images-01"></div>  
<div class="Content-Images-02"></div>  
<div class="Content-Images-03"></div>  
<div class="Content-Images-04"></div>
```

استفاده از تصاویر Data URIs

یکی دیگر از روش‌های کاهش درخواست‌های HTTP در یک سایت استفاده از [Data URIs](#) می‌باشد، توسط این روش می‌توانید فایل هایتان را درون HTML و یا CSS قرار دهید یا به اصطلاح embed کنید. به طور مثال جهت استفاده از یک تصویر می‌توانید به راحتی با آدرس دهی تصویر درون تگ img، تصویر را درون صفحه نمایش دهید:

```

```

همین کار را می‌توانیم توسط Data URIs انجام دهیم:

```

```

در کد فوق تصویر موردنظر را درون HTML به صورت embed شده قرار داده ایم، در این حالت دیگری نیازی به رفت و برگشت به سرور جهت نمایش تصویر نیست. **سینتکس Data URIs** به طور مثال تگ زیر را در نظر داشته باشید:

```

```

مقدار ویژگی src شامل موارد زیر است:

data: نام schema

image/png: نوع محتوا (content type)

base64: نوع encoding استفاده شده برای encode کردن اطلاعات

iVBOR...: اطلاعات encode شده.

توسط افزونه Web Essentials به راحتی می‌توانید تصویر موردنظرتان را به صورت Data URI تهیه کنید:

