

NuGet چیست؟

روش متداول استفاده از کتابخانه‌های موجود دات نتی در Visual studio عموماً به این صورت است: مراجعه به سایت مربوطه، دریافت بسته مورد نظر، باز کردن آن و سپس افزودن ارجاعی به اسمبلی‌های آن کتابخانه. در این حالت زمانیکه نسخه‌ی جدیدی از کتابخانه‌ی مورد استفاده ارائه شود (و عموماً تا مدت‌ها شاید از آن بی‌اطلاع باشیم) تمام این مراحل باید از ابتدا تکرار شوند و همینطور الی آخر.

برای رفع این نقیصه، تیم ASP.NET، افزونه‌ای سورس باز و رایگان را به نام NuGet جهت VS.Net 2010 طراحی کرده‌اند که کار مدیریت بسته‌های کتابخانه‌های مورد استفاده را بسیار ساده کرده است. امکانات این افزونه پس از نصب، در دو حالت استفاده از رابط گرافیکی کاربری آن و یا با استفاده از خط فرمان PowerShell ویندوز در دسترس خواهد بود. این افزونه در زمان بارگذاری، با مراجعه به فید سایت مرکزی خود، لیست بسته‌های مهیا را در اختیار علاقمندان قرار می‌دهد. در این سایت مرکزی به روی تمام توسعه دهنده‌ها جهت افزودن بسته‌های خود باز است.

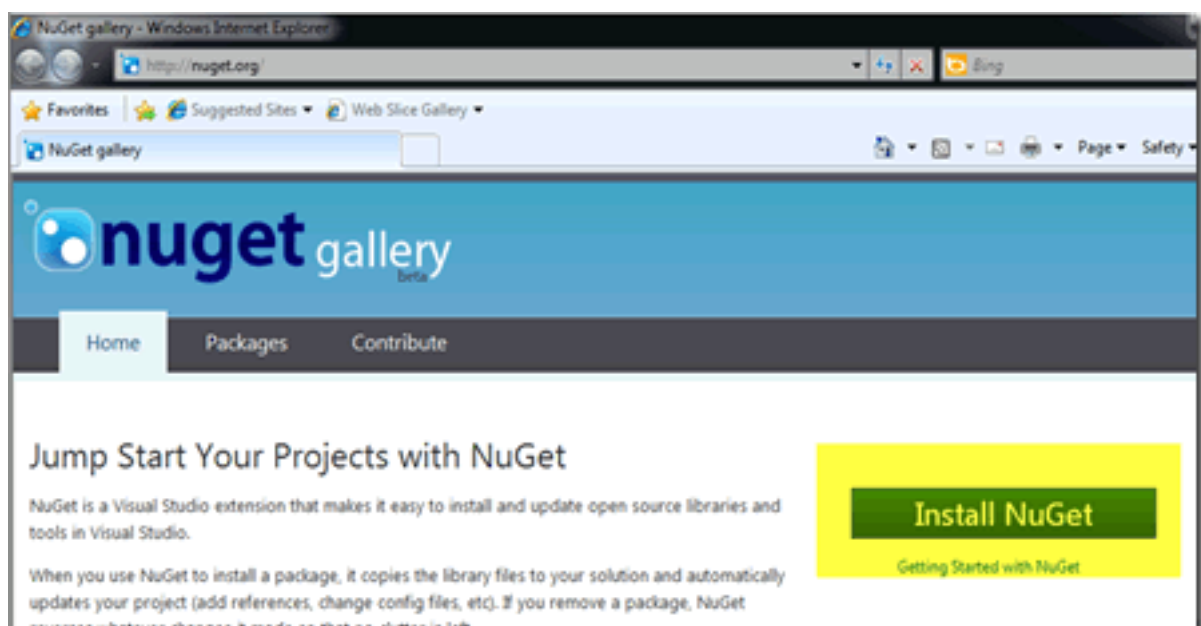
و ... فراگیری کار با NuGet برای تمام برنامه نویسان دات نت لازم و ضروری است! از این جهت که پیغام "این بسته تنها برای NuGet عرضه شده است" کم‌کم در حال متداول شدن می‌باشد و دیگر سایت‌های مرتبط، لینک مستقیمی را جهت دریافت کتابخانه‌های خود ارائه نمی‌دهند. حتی خبر به روز شدن محصولات خود را هم شاید دیگر به صورت منظم ارائه ندهند؛ زیرا NuGet کار مدیریت آن‌ها را به عهده خواهد داشت.

دریافت و نصب NuGet

NuGet را حداقل به سه طریق می‌توان دریافت و نصب کرد:

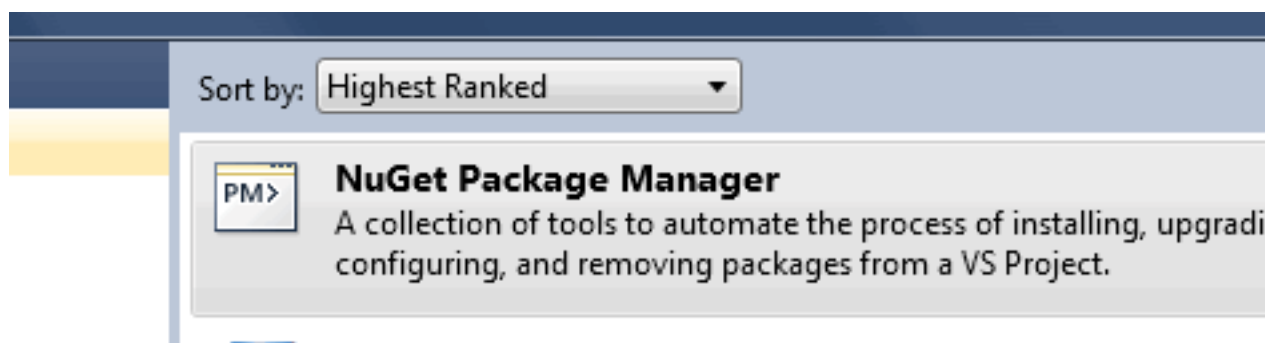
الف) با مراجعه به سایت CodePlex ([+](#))

ب) دریافت آن از سایت گالری‌های آن ([+](#))



(ج) با استفاده از امکانات VS.NET

هر سه روش فوق به دریافت و نصب فایل NuGet.Tools.vsix منتهی می‌شوند. برای مثال در روش (ج) باید به منوی Tools و گزینه‌ی Extension Manager مراجعه کنید. سپس برگه‌ی Online Gallery را گشوده و اندکی صبر کنید تا اطلاعات آن دریافت و نمایش داده شود. سپس NuGet را در Search box بالای صفحه نوشته و NuGet Package manager ظاهر شده را انتخاب و نصب کنید.

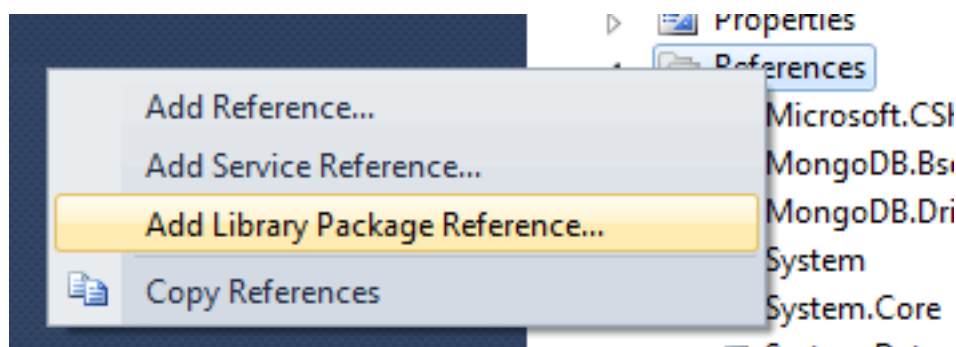


نحوه استفاده از NuGet

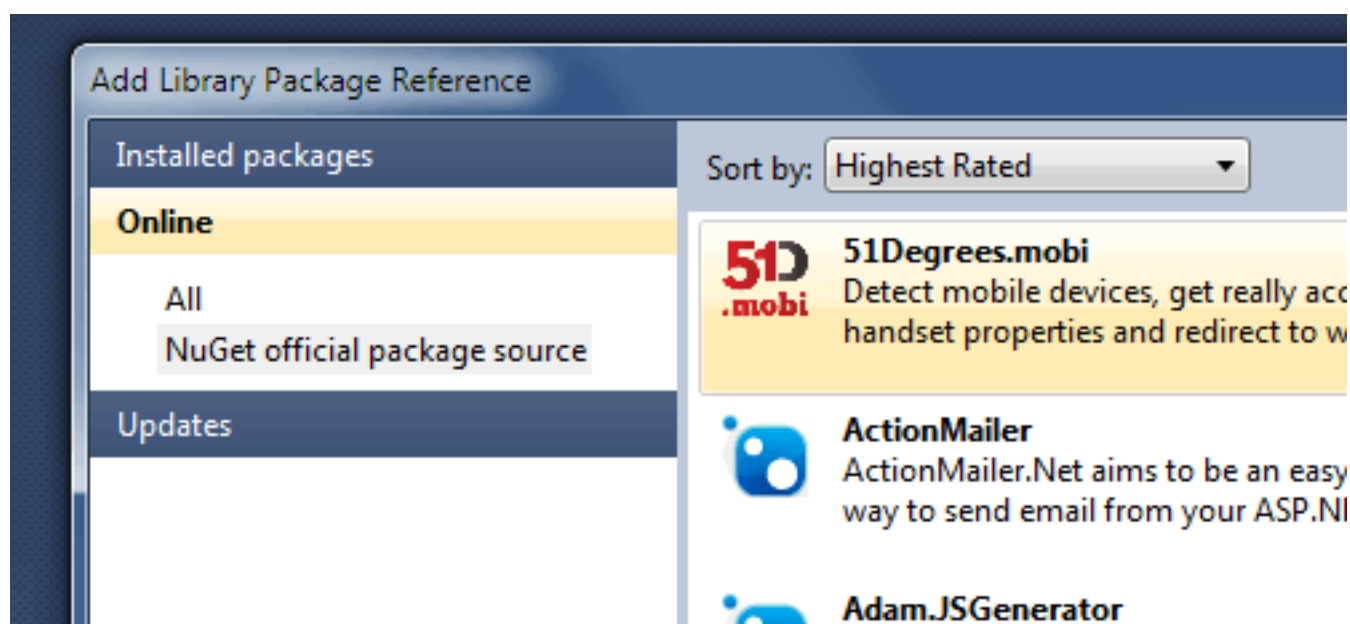
فرض کنید یک پروژه جدید ASP.NET را ایجاد کرده‌اید و نیاز است تا کتابخانه‌ی [ELMAH](#) به آن اضافه شود. روش انجام اینکار را به کمک NuGet در ادامه بررسی خواهیم کرد (کمتر از یک دقیقه زمان خواهد برد):

الف) با کمک امکانات رابط گرافیکی کاربر آن

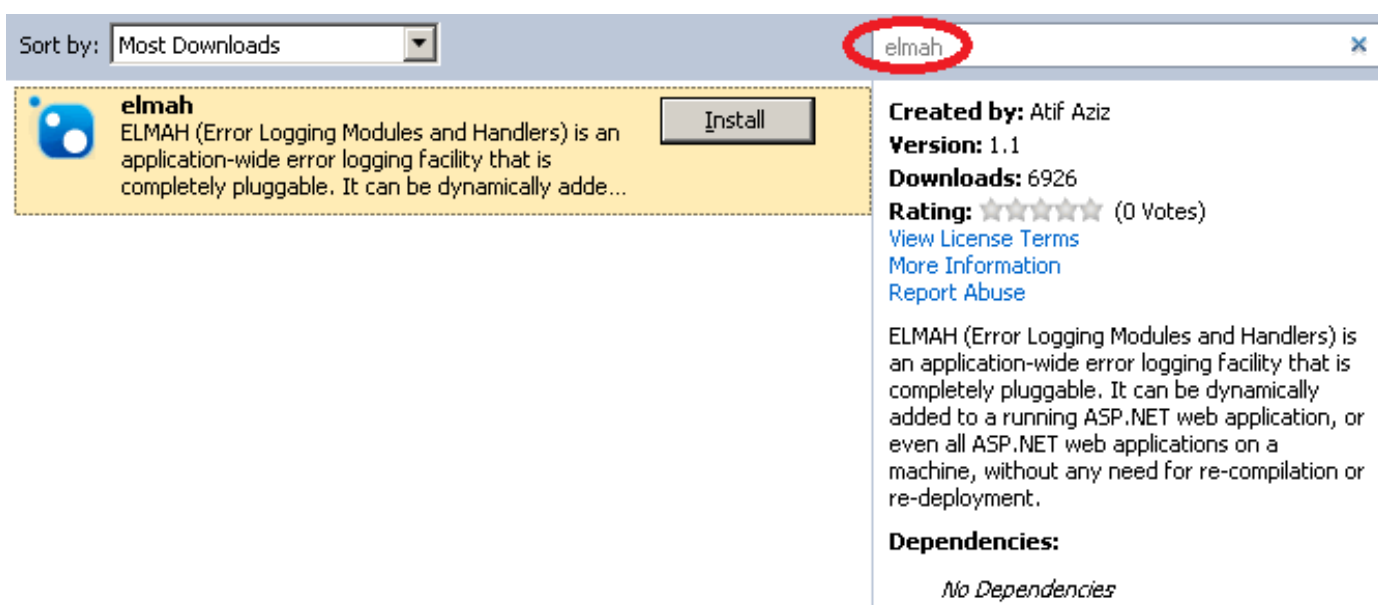
ساده‌ترین روش استفاده از NuGet، کلیک راست بر روی پوشه References در Solution explorer و سپس انتخاب گزینه‌ی Add Library Package Reference می‌باشد:



در صفحه‌ی باز شده، برگه‌ی Online را باز کنید و مدتی صبر نمایید تا اطلاعات لازم دریافت گردد (در زمان نگارش این مطلب، 1135 بسته در [این مخزن](#) موجود است):

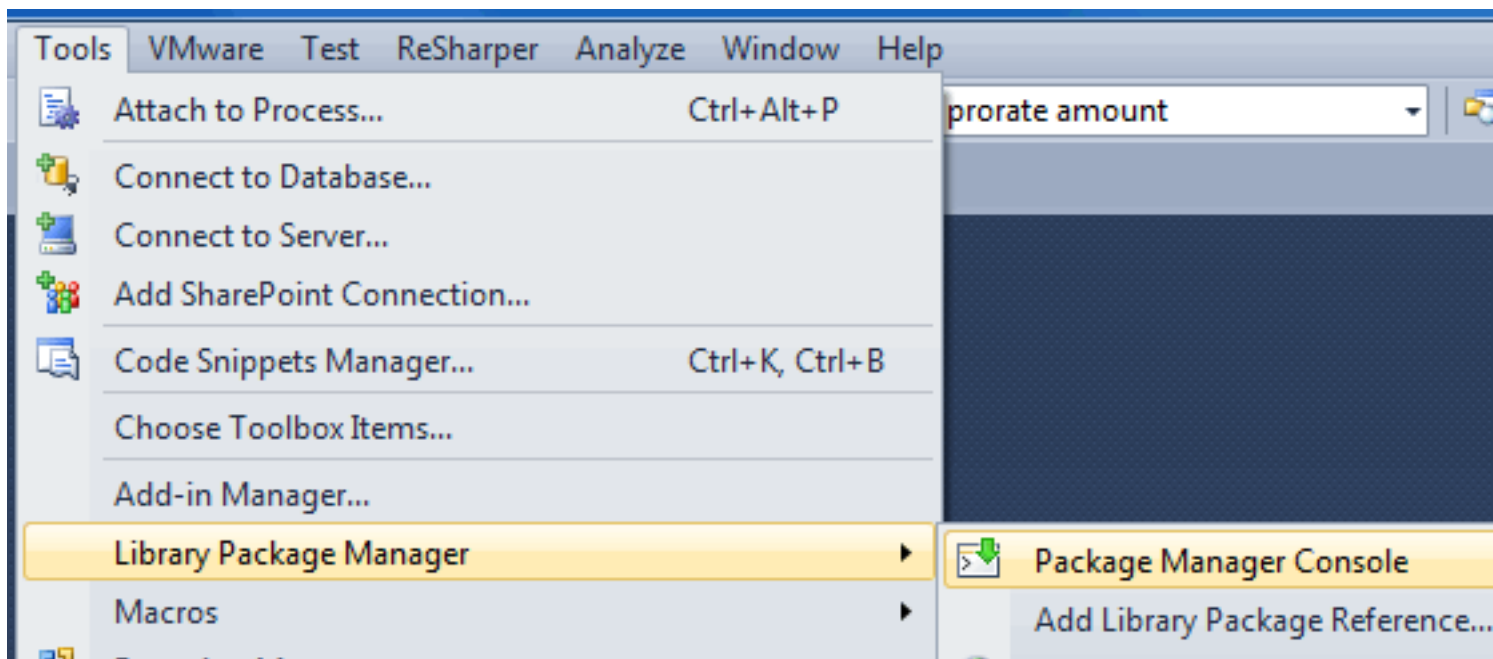


سپس در جعبه‌ی جستجوی سمت راست بالای صفحه، نام کتابخانه‌ی مورد نظر را نوشته و اندکی صبر کنید تا اطلاعات آن نمایش داده شود:

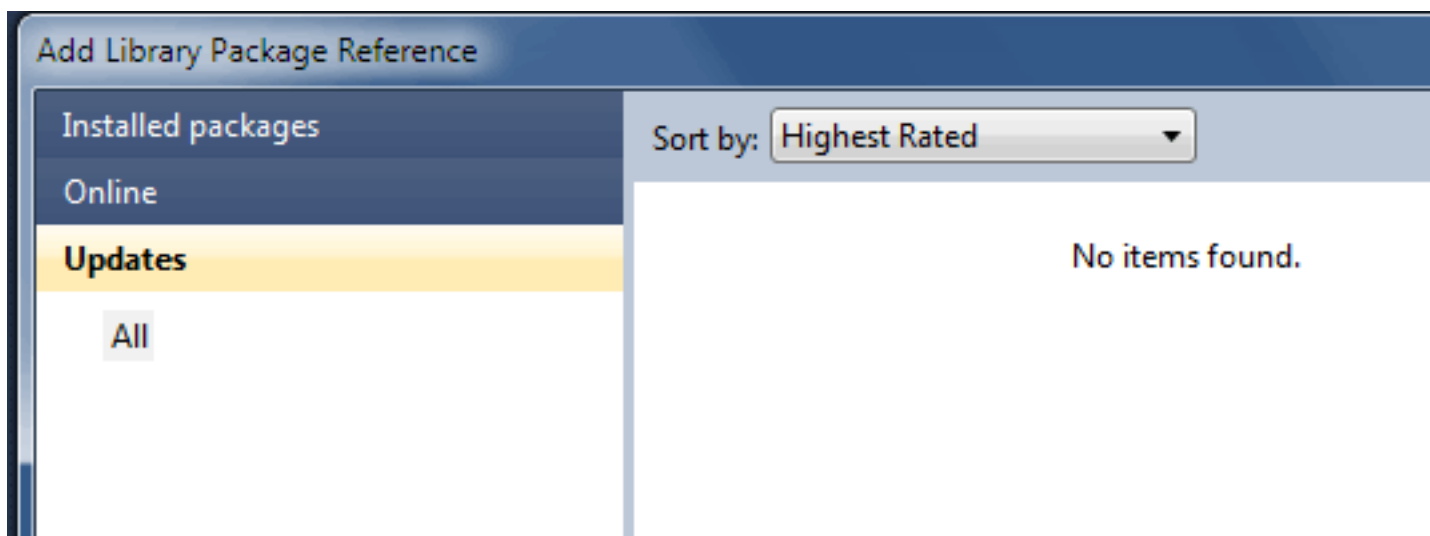


اکنون با کلیک بر روی دکمه Install، بسته مرتبط با این کتابخانه دریافت شده و سپس به صورت خودکار ارجاعی به آن نیز افزوده خواهد شد. همچنین تنظیمات مرتبط با فایل Config برنامه هم اضافه می‌شوند.

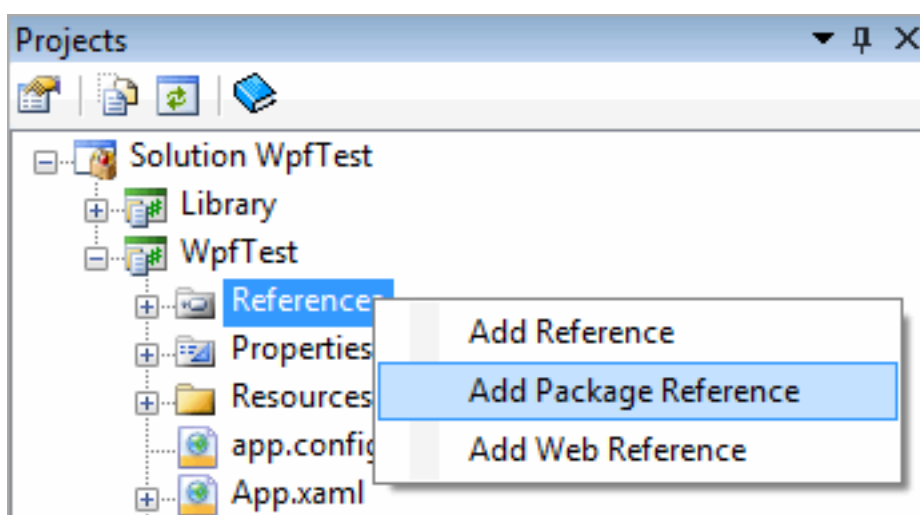
روش دیگر ظاهر کردن این صفحه، مراجعه به منوی Tools و گزینه‌ی Library Package Manager می‌باشد:



جهت دریافت به روز رسانی‌های بسته‌های نصب شده تنها کافی است به برگه‌ی Updates این صفحه مراجعه کرده و موارد لیست شده را نصب نمائیم:

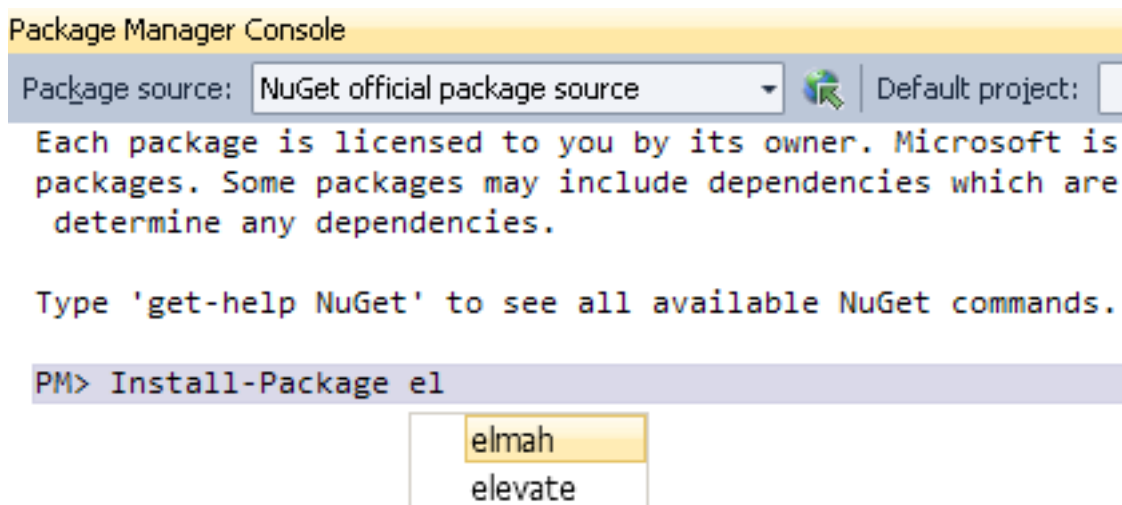


نکته: در NuGet [SharpDevelop 4.1](#) به بعد هم پشتیبانی می‌شود:



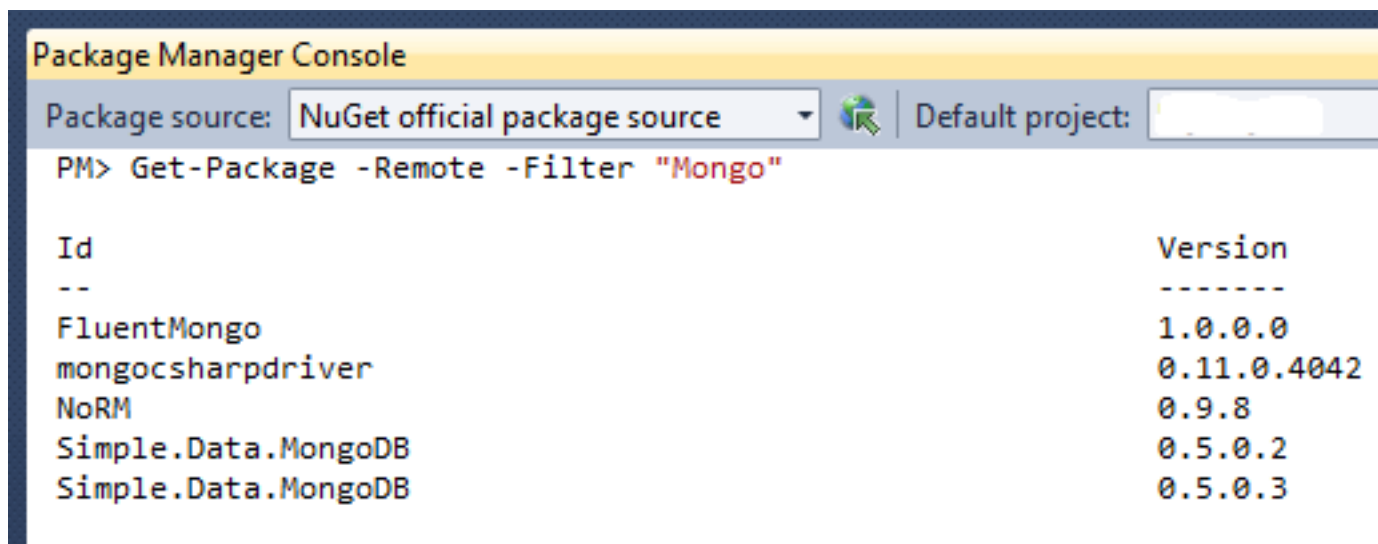
ب) با استفاده از امکانات خط فرمان PowerShell ویندوز برای استفاده از امکانات پاورشل ویندوز نیاز است تا [پاورشل نگارش 2](#) بر روی سیستم شما نصب باشد (نیاز به Windows XP with Service Pack 3 به بعد دارد). سپس به منوی Tools ، قسمت Library Package Manager ، گزینه‌ی Package Manager

Console آن جهت فعال سازی کنسول پاور شل در VS.NET مراجعه نمائید:



نکته: در تصویر فوق پس از نوشتن el ، دکمه tab فشرده شده است. در این حالت منوی پکیج‌های مهیای شروع شده با el ، از سایت مرکزی NuGet ظاهر گردیده است.

فرامین مهمی که در اینجا در دسترس هستند شامل: Get-Package ، List-Package ، Uninstall-Package ، Update-Package می‌باشند. برای مثال اگر قصد جستجو در بین بسته‌های موجود را داشته باشید Get-Package بسیار مفید است:



برای مثال جهت یافتن بسته‌های مرتبط با wpf و silverlight به صورت زیر می‌توان عمل کرد:

```
PM> get-package -remote -filter wpf
PM> get-package -remote -filter silverlight
```

نکته: روش دیگر جستجو در بین بسته‌های مهیا، مراجعه به سایت گالری آن است: (+). در اینجا دستور پاورشل نصب هر بسته‌ی یافت شده نیز نمایش داده می‌شود.

ج) استفاده از برنامه NuGet.exe
برنامه NuGet.exe از سایت CodePlex قابل دریافت است. این روش هم جهت علاقمندان به خط فرمان تدارک دیده شده است! پس از دریافت آن فرض کنید می‌خواهیم تمام بسته‌های شروع شده با nhi را لیست کنیم. برای این منظور دستور خط فرمان ذیل را صادر کنید:

```
D:\Test>nuget list nhi
```

سپس برای دریافت مثلا NHibernate تنها کافی است دستور زیر اجرا شود:

```
D:\Test>nuget install NHibernate
```

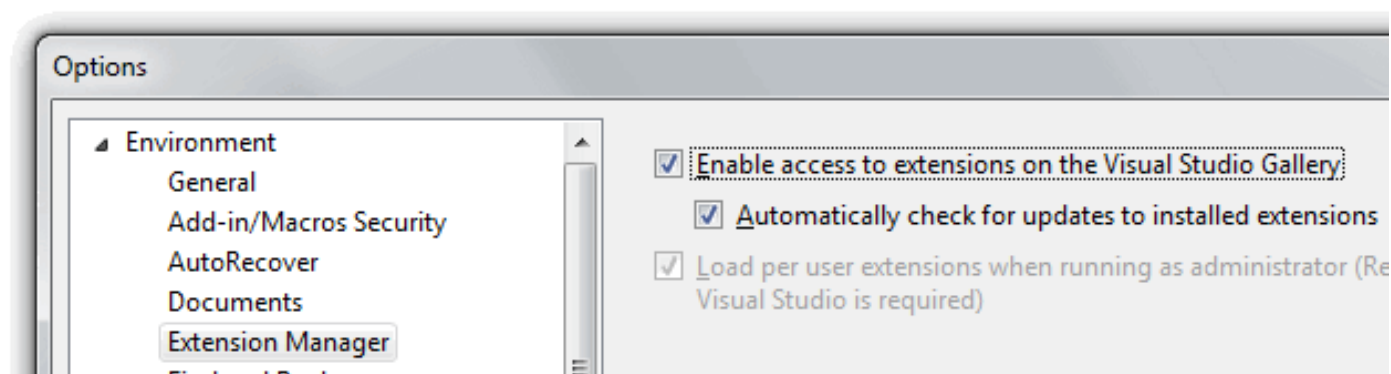
به این صورت کتابخانه NHibernate به همراه تمام وابستگی‌های آن دریافت خواهد شد.

به روز رسانی خودکار NuGet
برای به روز رسانی برنامه nuget.exe دستور زیر را می‌توان صادر نمود:

```
D:\Test>NuGet.exe u
```

و یا جهت فعال سازی به روز رسانی‌های خودکار افزونه‌ها در VS.NET به منوی زیر مراجعه کنید:

Tools | Options, then Environment | Extension Manager and click "Automatically check for updates to installed extensions."



ادامه دارد ...

نظرات خوانندگان

نویسنده: علی اقدم
تاریخ: ۲۲:۴۸:۱۲ ۱۳۹۰/۰۱/۲۱

بسیار مفید است و اگر ادامه پیدا کند خواهد بود؛)

برام یه تلنگر بود برم سمت NuGet

راستی چطور تلفظ میشه؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۵۳:۳۰ ۱۳۹۰/۰۱/۲۱

داستان نامگذاری اون رو می‌تونید اینجا بخونید

<http://haacked.com/archive/2010/10/29/nupack-is-now-nuget.aspx>

نیو گت است مطابق نظر تولید کنندگان آن

نویسنده: afsharm
تاریخ: ۱۷:۰۴:۳۳ ۱۳۹۰/۰۱/۲۲

سلام،

یک سوال بی‌ربط به NuGet:

disqus در وبلاگ من به خوبی کار نمی‌کند. در واقع box مربوط به ورود کامنت مدت زمان خیلی خیلی زیادی به حالت Please wait می‌ماند. آن هم ظاهراً به خاطر بلایی که بر سر mediacdndisqus.com آمده.

ایده‌ای برای حل آن دارید؟

افشار

blog.afsharm.com

نویسنده: وحید نصیری
تاریخ: ۱۹:۲۶:۵۹ ۱۳۹۰/۰۱/۲۲

سلام؛ من با قسمت نظرات سایت شما مشکلی نداشتم. الان با دو مرورگر مختلف بررسی کردم و به خوبی دیسکاس نمایش داده میشه.

ضمناً دیسکاس زبان فارسی رو هم اضافه کرده که می‌تونید فعالش کنید

Admin --- Settings --- General --- Language --- Persian

نویسنده: afsharm
تاریخ: ۲۲:۱۰:۳۷ ۱۳۹۰/۰۱/۲۲

ممنون.

جالبه که درست در همون لحظه‌ای که کامنت قبلی رو گذاشتم سیستم کامنت وبلاگ خودم هنوز مشکل داشت ولی همین الان هیچ مشکلی نداره!

نویسنده: علیرضا

تاریخ: ۱۴:۲۸ ۱۳۹۲/۱۰/۲۹

با سلام و خسته نباشید
من یه مشکل کوچیک دارم تو نوگت با ویژوال استدیو 2012 update 2 یا ویژوال استدیو 2013
به آدرس https در package source وصل نمیشه (تا اونجایی که فهمیدم به خاطر سیاست‌های داخلی اینترنت ایران)، میدونم با
تغییر دادن آدرس به http مشکل حل میشه!
ولی متأسفانه الان هر چی میگردم اون آدرس رو پیدا نمیکنم
خواستم ببینم اگر اون آدرس package source رو دارید برام قرار بدید
با تشکر از شما

نویسنده: وحید نصیری

تاریخ: ۱۴:۳۵ ۱۳۹۲/۱۰/۲۹

« [خلاصه‌ای در مورد روش‌های دریافت فایل از سایت NuGet](#) »

عنوان: آشنایی با NuGet - قسمت دوم

نویسنده: وحید نصیری

تاریخ: ۱۸:۵۹:۰۰ ۱۳۹۰/۰۱/۲۳

آدرس: www.dotnettips.info

برچسب‌ها: NuGet

قسمت قبل

از دید یک مصرف کننده بود؛ این قسمت جهت توسعه دهنده‌ها تهیه شده است. کسانی که قصد دارند تا بسته‌های NuGet ایی از کارشان تهیه کنند. مراحل اینکار به شرح زیر است:

الف) برای این منظور نیاز است تا برنامه‌ی خط فرمان NuGet.exe معرفی شده در قسمت قبل را ابتدا دریافت کنید: (+)

ب) برای بسته نرم افزاری خود یک پوشه جدید درست کنید. سپس فرمان `nuget.exe spec` را در این پوشه صادر نمایید. بلافاصله فایل به نام `Package.nuspec` تشکیل خواهد شد:

```
D:\Prog\1389\CodePlex\slpdatepicker\SLPDatePickerNuGet>NuGet.exe spec
Created 'Package.nuspec' successfully.
```

فایل `Package.nuspec`، یک فایل XML ساده است. آنرا با یک ادیتور متنی باز کرده و تغییرات لازم را اعمال نمایید. برای مثال من جهت پروژه [Silverlight 4 Persian DatePicker](#)، محتویات آنرا به صورت زیر تغییر داده‌ام:

```
<?xml version="1.0"?>
<package xmlns="http://schemas.microsoft.com/packaging/2010/07/nuspec.xsd">
  <metadata>
    <id>Silverlight.4.Persian.DatePicker</id>
    <version>1.0</version>
    <authors>Vahid Nasiri</authors>
    <owners>Vahid Nasiri</owners>
    <licenseUrl>http://slpdatepicker.codeplex.com/license</licenseUrl>
    <projectUrl>http://slpdatepicker.codeplex.com/</projectUrl>

    <iconUrl>https://slpdatepicker.svn.codeplex.com/svn/SilverlightPersianDatePicker/Views/Images/date.png</iconUrl>
    <requireLicenseAcceptance>false</requireLicenseAcceptance>
    <description>Silverlight 4 Persian DatePicker Control</description>
    <tags>Silverlight WPF Persian DatePicker</tags>
  </metadata>
  <files>
    <file src="..\SilverlightPersianDatePicker\Bin\Release\*.dll" target="lib" />
    <file src="..\SilverlightPersianDatePicker\Bin\Release\*.pdb" target="lib" />
    <file src="..\SilverlightPersianDatePicker\Bin\Release\*.xml" target="lib" />
  </files>
</package>
```

همانطور که ملاحظه می‌کنید یک سری اطلاعات عمومی از پروژه مورد نظر درخواست شده است؛ برای مثال آدرس آیکن آن چیست یا کجا می‌توان آنرا یافت؟ مجوز استفاده از آن چیست و مواردی از این دست. به کمک تگ `files` هم فایل‌های کتابخانه در اینجا لحاظ شده‌اند. فایل آیکن معرفی شده باید در اندازه‌ی 32*32 و با فرمت `png` باشد. باید دقت داشت که در سایت `nuget.org`، بسته شما بر اساس `id` ذکر شده معرفی خواهد شد و آدرسی بر این اساس تشکیل می‌گردد. بنابراین از فاصله یا موارد مشکل ساز در این بین استفاده نکنید.

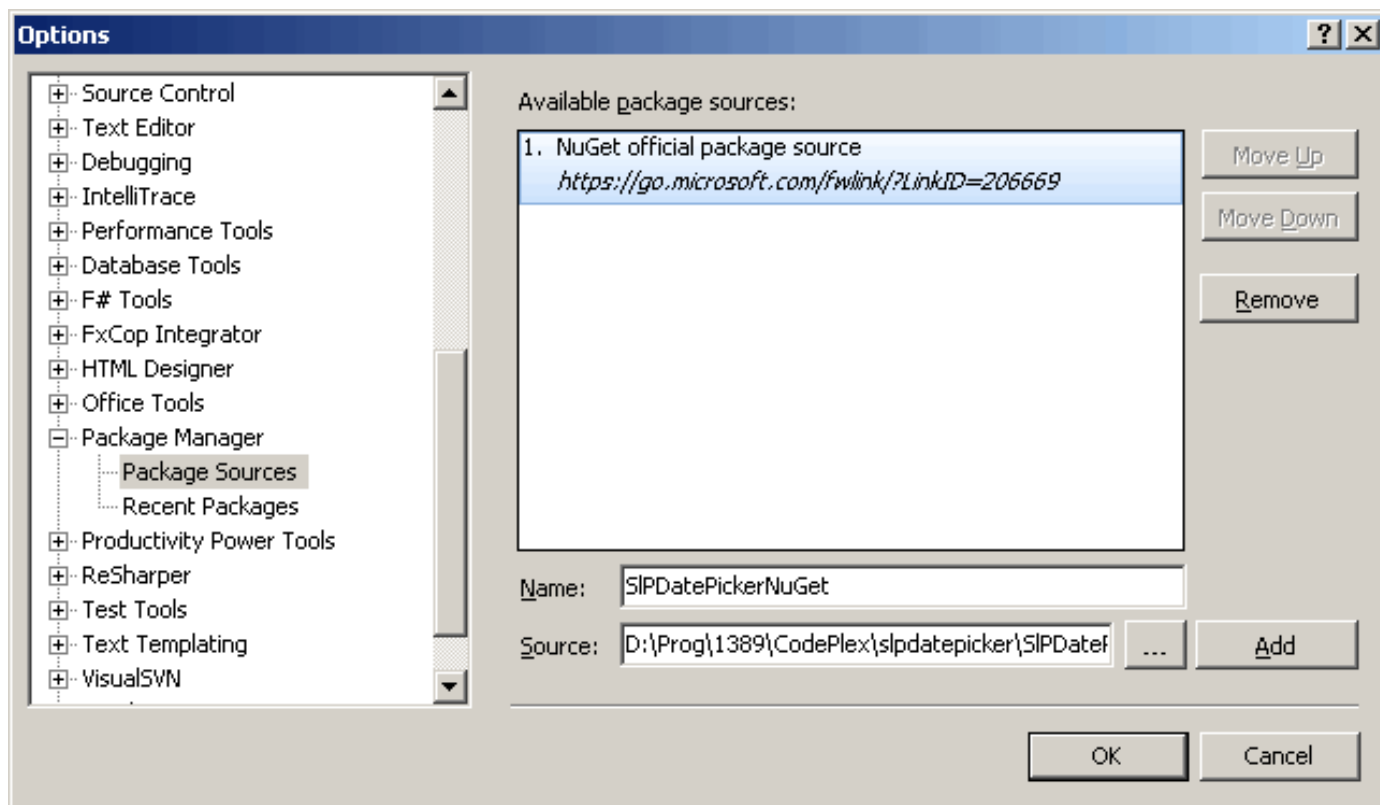
در مورد نحوه‌ی ایجاد قدم به قدم یک پروژه جدید در سایت کدپلکس می‌توان به این مطلب مراجعه نمود: (+)

ج) اکنون نوبت به تهیه بسته نهایی می‌رسد. برای این منظور دستور زیر را در خط فرمان صادر کنید:

```
NuGet.exe pack Package.nuspec
```

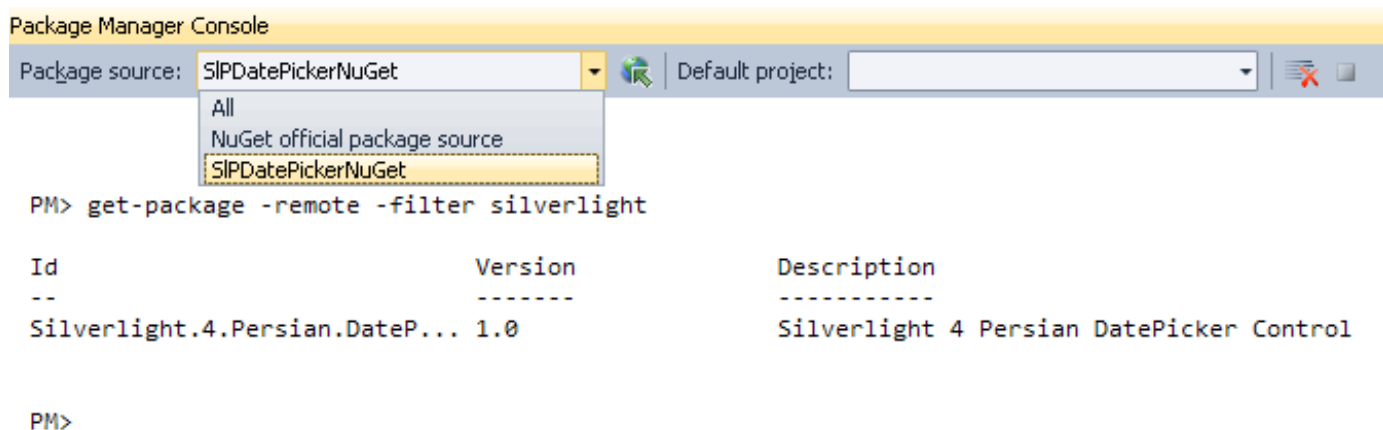
پس از چند لحظه فایل Silverlight.4.Persian.DatePicker.1.0.nupkg جهت ارائه عمومی تولید خواهد شد.

(د) قبل از اینکه این فایل نهایی را در سایت nuget.org آپلود کنیم، می‌توان مشخصات آن را به صورت محلی نیز یکبار مرور کرد. برای این منظور در VS.NET به منوی Tools گزینه‌ی Options مراجعه کرده و در قسمت package manager، آدرس پوشه بسته مورد نظر را وارد کنید. برای مثال:

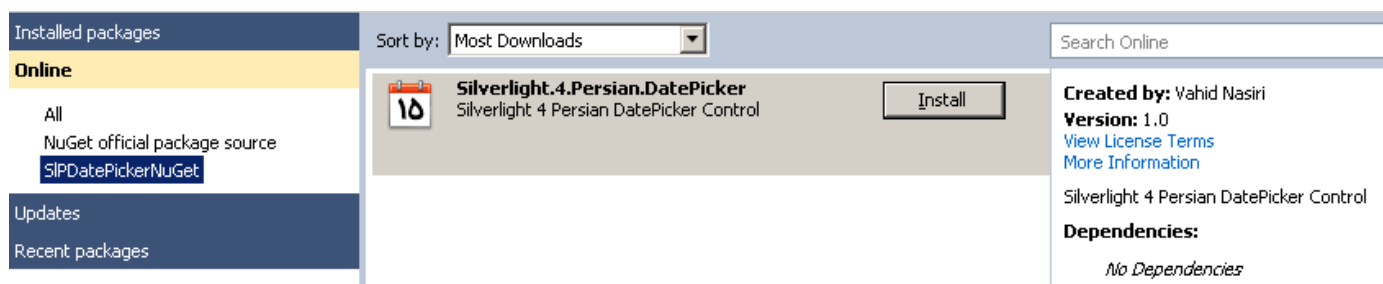


اکنون اگر کنسول پاورشل توضیح داده شده در قسمت قبل را باز نمائید، منبع جدید اضافه شده مشخص است یا می‌توان توسط دستور ذیل از آن کوئری گرفت:

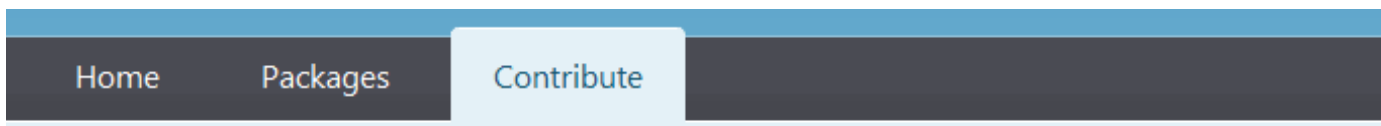
```
get-package -remote -filter silverlight
```



و یا اگر همانند توضیحات قبل به صفحه‌ی دیالوگ add library package reference مراجعه کنیم، مشخصات کامل بسته به همراه منبع محلی باید قابل مشاهده باشند:



۵) پس از بررسی محلی بسته مورد نظر، اکنون نوبت به ارائه عمومی آن می‌باشد. برای این منظور ابتدا باید در سایت nuget.org ثبت نام کرد: (+). اگر آدرس ایمیل شما را نپذیرفت، از مرورگر IE استفاده کنید! پس از ثبت نام تنها کافی است به قسمت contribute سایت مراجعه کرده و فایل بسته نهایی را در آنجا آپلود کرد. به این صورت بسته نهایی در سایت پدیدار خواهد شد: (+) همچنین بلافاصله در قسمت گالری آنلاین صفحه add library package reference نیز قابل دسترسی خواهد بود.



New Contribution

1



Submit your package for others to enjoy.

- ☒ Upload a package file from my local computer
- ☐ Submit a URL to a package file located on my web site

Your package file will be up

در آینده جهت توزیع به روز رسانی‌های جدید، همین مراحل باید تکرار شوند. البته در نظر داشته باشید که version ذکر شده در فایل Package.nuspec را باید حتما تغییر داد تا بسته‌ها از یکدیگر متمایز شوند. امکان اتوماسیون این توزیع نیز وجود دارد. همان فایل nuget.exe ، امکان ارسال بسته نهایی را به سایت nuget.org نیز دارد:

```
nuget push name.nupkg key
```

در اینجا key مخصوص به خود را می‌توان در صفحه‌ی <http://nuget.org/Contribute/MyAccount> مشاهده و استفاده نمود.

اگر علاقمند به مشاهده جزئیات بیشتری از این پروسه هستید، می‌توان به سایت رسمی آن مراجعه کرد: ([+](#))

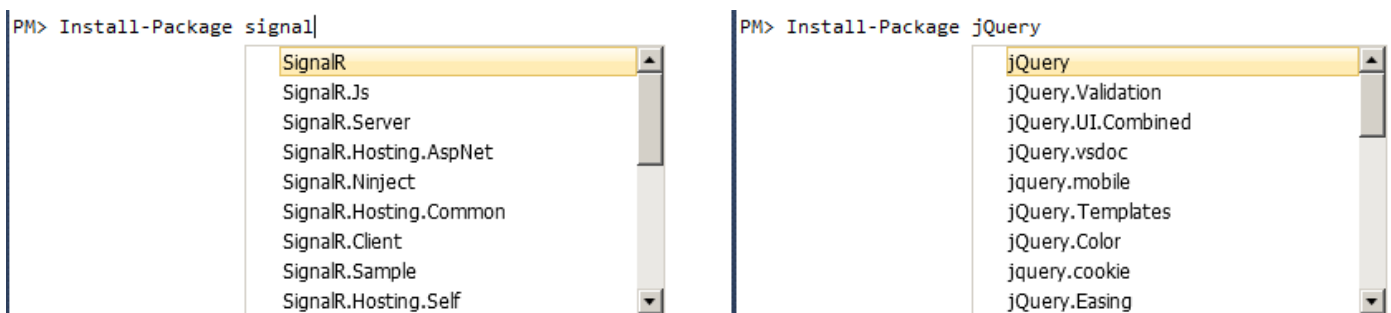
عنوان: **NuGet 2.0 منتشر شد**
نویسنده: یوسف نژاد
تاریخ: ۱۴۰۲/۰۴/۱۳
آدرس: www.dotnettips.info
برچسب‌ها: NuGet

نسخه جدید برنامه مدیریت بسته‌های دات نت هم آماده شد. میتونین از [اینجا](#) دانلودش کنین. (2.5MB)

طبق آمار خود سایت [نوگت](#) تا حالا بیش از 14.6 میلیون بار بسته‌های اون توسط کاربران دانلود شدن و بیش از 7000 بسته متمایز در این گالری موجوده!

[NuGet 2.0 Release Notes](#)

یکی از مشکلاتی که تو این نسخه رفع شده موردی بود که در ارتباطات کند اینترنت بوجود میومد (^). تو کنسول این افزونه پس از وارد کردن قسمتی از نام یک بسته با فشردن کلید TAB نام تمام بسته‌هایی که اول اسمشون اون عبارت تایپ شده باشه، لیست میشه (Tab Completion).



این عملیات در نسخه‌های قبلی با استفاده از یک درخواست HTML به [OData](#) (^) انجام میشد که داده‌هایی بیش از حد نیاز رو برمیگردوند. اما تو این نسخه این عملیات با استفاده از یک درخواست سریع JSON انجام میشه. (^)

من خودم این ویژگی رو تست کردم و افزایش سرعتش قابل ملاحظه بود!

یکی دیگه بهبودهای حاصله در دریافت بسته‌های وابسته نسبت به نسخه دات نت فریمورک در پروژه هدف هستش. یعنی میشه تنظیماتی روی بسته‌های نوگت اعمال کرد تا به صورت هوشمندانه نسخه متناسب با دات نت فریمورک مورد استفاده رو دانلود و نصب کنه.

اگه به صفحه مربوط به مشکلات و درخواستهای کاربران برای این افزونه (^) مراجعه کنین میبینید این دو مورد که تو این نسخه برطرف شدن بیشترین تقاضا رو داشتن. مورد اول با توجه به سرعت پایین اینترنت برای خود من خیلی کاربردی.

نکته : اگر از VS 2010 SP1 استفاده میکنین هنگام به روز رسانی نوگت با استفاده از Extension Manager خود VS ممکنه با خطا مواجه بشین. اصولاً بهترین روش نصب نوگت ابتدا uninstall کردن نسخه قدیمی این افزونه از طریق appwiz.cpl و سپس نصب نسخه جدید با اجرای فایل vsix هست (در هر دو قسمت نیاز به دسترسی administrator دارین).

و در نهایت:

(^). You can develop your own package and share it via the NuGet Gallery

ویرایش:

متأسفانه دیروز وقت نکردم نتایج آزمایش با [Fiddler](#) رو اینجا بزارم.

مورد اول (Tab Completion) رو با سه نسخه از نوگت (1.6 و 1.8 و 2.0) برای دو عبارت jq و signa تست کردم و نتایج بدست آمده به شرح زیره. در زیر url درخواست مربوطه به همراه حجم دیتای دریافتی آورده شده:

نوگت 1.6:

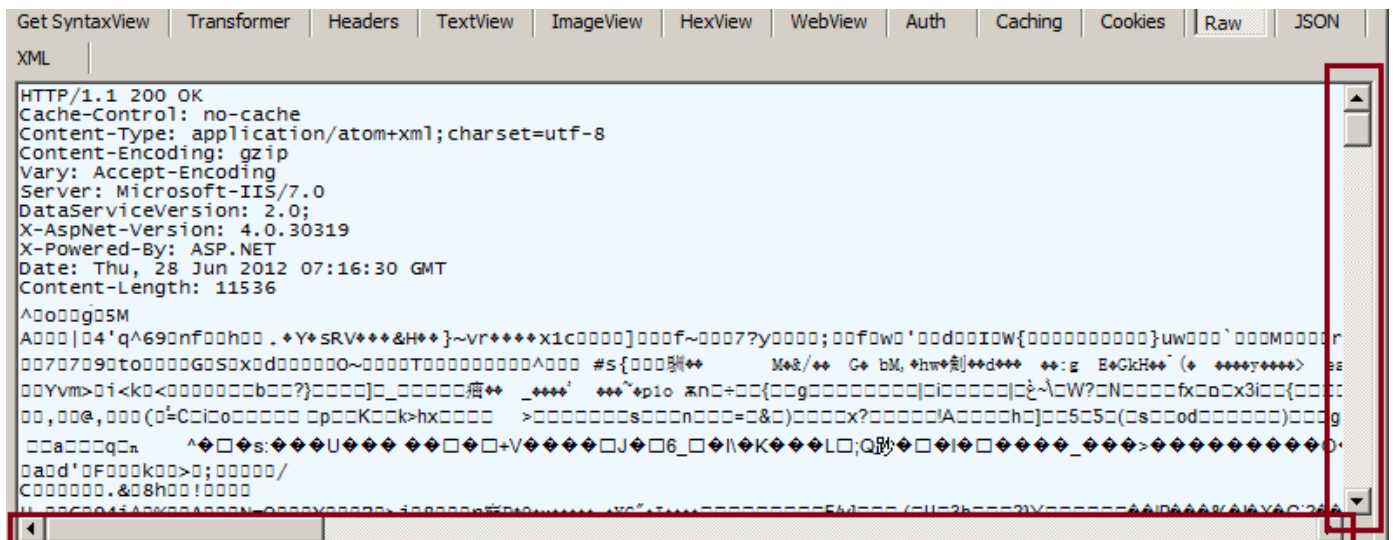
```

nuget ver 1.6

jq:
GET
https://nuget.org/api/v2/Packages()?orderby=DownloadCount%20desc,Id&$filter=startswith(tolower(Id),'jq
ue')%20and%20IsLatestVersion&$skip=0&$top=90 HTTP/1.1
53691 bytes

signa:
GET
https://nuget.org/api/v2/Packages()?orderby=DownloadCount%20desc,Id&$filter=startswith(tolower(Id),'si
gna')%20and%20IsLatestVersion&$skip=0&$top=90 HTTP/1.1
11536 bytes

```



نوگت 1.8:

```

nuget ver 1.8

jq:
GET
https://nuget.org/api/v2/Packages()?orderby=DownloadCount%20desc,Id&$filter=startswith(tolower(Id),'jq
ue')%20and%20IsLatestVersion&$skip=0&$top=90 HTTP/1.1
53694 bytes

signa:

```

```
GET
https://nuget.org/api/v2/Packages()?orderby=DownloadCount%20desc,Id&$filter=startswith(tolower(Id),'si
gna')%20and%20IsLatestVersion&$skip=0&$top=90 HTTP/1.1
11536 bytes
```

6	200	HTTPS	nuget.org	/api/v2/Packages()?orderby=DownloadCount%20desc,Id&\$filter=startswith(tolower(Id),'jqe')%2...	53,694
7	200	HTTPS	nuget.org	/api/v2/Packages()?orderby=DownloadCount%20desc,Id&\$filter=startswith(tolower(Id),'signa')%...	11,536

نوگت 2.0:

nuget ver 2.0

```
jqe:
GET https://nuget.org/api/v2/package-ids?partialId=jqe HTTP/1.0
598 bytes
```

```
signa:
GET https://nuget.org/api/v2/package-ids?partialId=signa HTTP/1.0
457 bytes
```

پاسخ سرور به عبارت signa در نسخه 2.0:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.0
X-AspNetMvc-Version: 3.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Thu, 28 Jun 2012 07:23:06 GMT
Connection: close
Content-Length: 457
```

```
["SignalR","SignalR.Client","SignalR.Client.Silverlight","SignalR.Client.Silverlight5","SignalR.Client.
WP7","SignalR.EventStream","SignalR.Hosting.AspNet","SignalR.Hosting.Common","SignalR.Hosting.Owin","Si
gnalR.Hosting.Self","SignalR.Js","SignalR.MicroSliver","SignalR.Ninject","SignalR.RabbitMq","SignalR.Re
active","SignalR.Redis","SignalR.Sample","SignalR.Server","SignalR.StructureMap","SignalR.WebSockets","
SignalR.WindowsAzureServiceBus","Signals.js"]
```

میبینید که غیرقابل مقایسه هستن! و همونطور که آقای هنسلمن ([^](#)) در مورد نسخه‌های قدیمی گفتن: « داده‌های بیش از حد نیاز رو برمیگردونن »

نسخه‌های قبل از 2.0 همگی از odata استفاده میکردن و داده‌های برگشتی تماما encrypt شده‌اند. اما در نسخه 2.0 داده‌های برگشتی از نوع JSON هست (به Fiddler مراجعه کنین تا تفاوت واقعی رو ببینین. البته برای اینکه داده‌های encrypt شده رو در fiddler ببینین باید طبق راهنمایی خودش از تنظیمات مربوط به decrypt ترافیک https استفاده کنین)

نظرات خوانندگان

نویسنده: علیرضا صالحی
تاریخ: ۲۲:۱۱/۰۴/۱۳۹۱

اگر موقعی که نسخه قدیمی Nuget نصب باشه اقدام به نصب ASP.NET MVC 4 RC کنید با شکست مواجه خواهید شد هیچ راهی هم جز حذف کردن Nuget ندارید.

بسته MVC خودش حاوی Nuget هست و نمیدونم چرا نمیتونه آپدیتش کنه!

نویسنده: یوسف نژاد
تاریخ: ۱۱:۲۹/۰۴/۱۳۹۱

درهر حالتی اگه شما بخواین نسخه ای از نوگت رو به روز رسانی بکنین یکسری خطاهایی ممکنه به وجود بیاد. البته ظاهرا این مشکل نوگت نیست ([^](#) و [^](#)) بلکه به یه ایرادهای خاص در Visual Studio Extension manager بر میگردد ([^](#) و [^](#)) : [Hotfix](#) و یا این مورد:

Attempting to install or uninstall results in the error "Cannot create a file when that file already exists."

For some reason, Visual Studio extensions can get in a weird state where you've uninstalled the VSIX extension, but some files were left behind. To work around this issue:

1. Exit Visual Studio
 2. Open the following folder (it might be on a different drive on your machine)
C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\IDE\Extensions\Microsoft Corporation\NuGet Package Manager\<version>\
 3. Delete all the files with the .deleteme extensions.
 4. Re-open Visual Studio
- After following these steps, you should be able to continue.

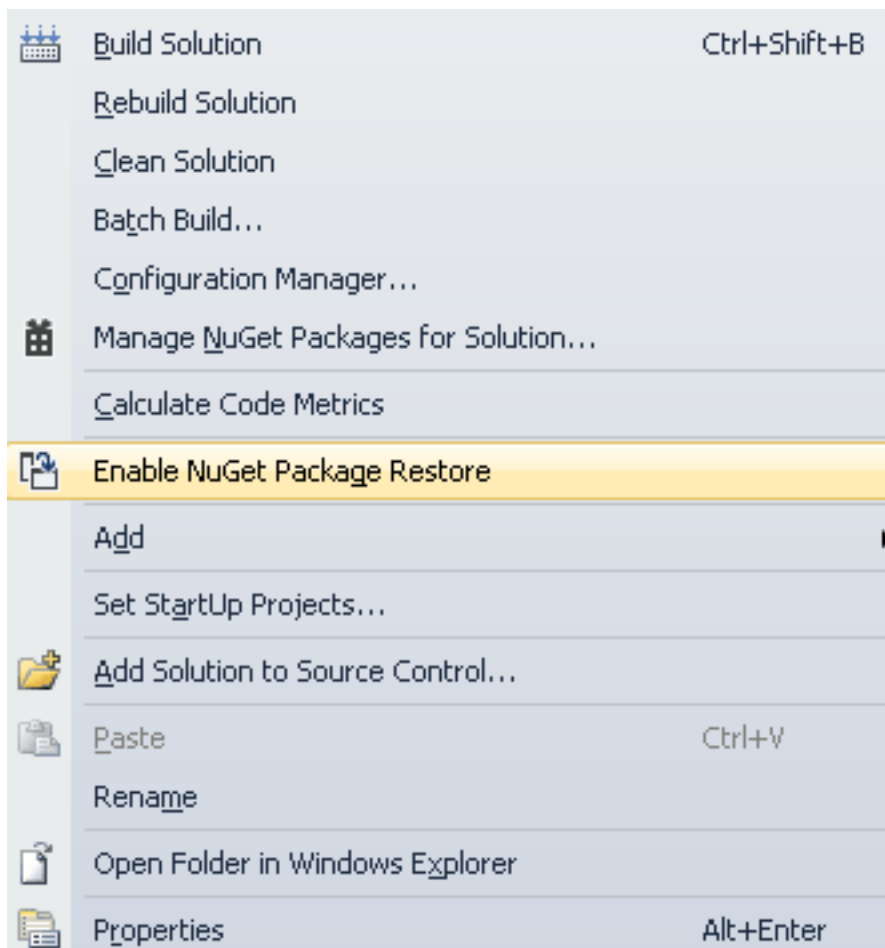
اگر قصد انتشار عمومی پروژه خود را دارید، نیازی به ارائه پوشه packages آن نیست. استفاده کننده نهایی به روشی که در ادامه توضیح داده خواهد شد، می‌تواند ارجاعات کل Solution را به یکباره به روز نماید؛ البته اگر تنها فایل یا فایل‌های packages.config پروژه‌های موجود پیوست شده و موجود باشند.

یک آزمایش

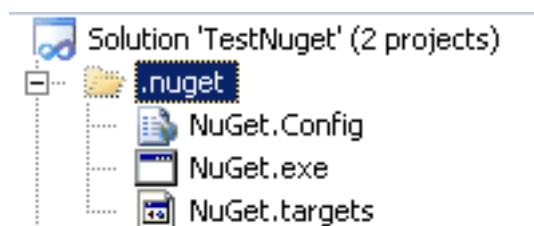
یک پروژه کنسول جدید را آغاز می‌کنیم. سپس به Solution آن یک Class library آزمایشی را نیز اضافه خواهیم کرد. اکنون در خط فرمان پاورشل نیوگت (Tools > Library Package Manager > Package Manager Console) به هر یک از این دو پروژه، ارجاعی را به بسته structuremap از طریق دستور زیر اضافه می‌کنیم:

```
Install-Package structuremap
```

اکنون یکبار پروژه را کامپایل کرده و سپس VS.NET را خاتمه می‌دهیم. در ادامه پوشه‌های packages و همچنین bin و obj را کلاً حذف می‌کنیم؛ اما فایل‌های متنی packages.config پروژه‌ها را نگه خواهیم داشت. مجدداً به VS.NET مراجعه خواهیم کرد. اینبار بر روی Solution کلیک راست کرده و گزینه «Enable NuGet Package Restore» را انتخاب می‌کنیم:



یک پوشه حاوی NuGet.exe به Solution جاری اضافه خواهد شد:



اکنون اگر پروژه را Build کنیم، تمام ارجاعات را به صورت خودکار از اینترنت (و یا کش موجود بر روی سیستم) دریافت و به Solution اضافه می‌کند.
به علاوه پوشه Packages نیز مجدداً بازسازی خواهد شد.

پس از اینکار نهایتاً برای اطمینان خاطر می‌توان دستور ذیل را در خط فرمان پاورشل نیوگت صادر کرد:

```
Update-Package -Safe
```

این دستور به یکباره کلیه ارجاعات موجود packages را بررسی کرده و به روز خواهد کرد. پارامتر safe آن اختیاری است. اگر عنوان شود، سعی می‌کند همان شماره نگارشی را که در فایل‌های packages.config موجود است، دریافت و به روز نماید. در غیر اینصورت، آخرین فایل و آخرین نگارش موجود را دریافت و به روز رسانی خواهد کرد. به این ترتیب می‌توان به صرفه جویی زمانی قابل توجهی در یک پروژه با ارجاعات زیاد، رسید.
برای نمونه دستور update-package را بر روی یک پروژه MVC4 اجرا کنید تا این صرفه جویی زمانی را بهتر بتوانید [حس کنید](#) !

نتیجه گیری

لطفاً حین ارائه عمومی پروژه خود، پوشه‌های bin، obj و همچنین packages آن را حذف کنید. استفاده کننده صرفاً با داشتن فایل‌های packages.config به کمک روشی که عنوان شد می‌تواند ارجاعات کل Solution را بازیابی کند.

نظرات خوانندگان

نویسنده: توحید عزیزی
تاریخ: ۱۳۹۲/۰۲/۲۸ ۰:۱۲

بسیار عالی. سپاسگزارم.

نویسنده: Ara
تاریخ: ۱۳۹۲/۰۲/۳۰ ۱۴:۵۳

سلام

خسته نباشید

می‌خواستم بدونم برای اینکه یک پکیج لوکالی استفاده بشه چه باید کرد !
مثلا مثلا DLL های Framework رو بشه Package کرد و همکاران شرکت پس تغییر یک dll بتونن اون رو تو سیستم خودشون آپدیت کنند !

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۳۰ ۱۵:۴

برای استفاده لوکال در شبکه داخلی، امکان تهیه یک NuGet Server لوکال وجود داره: « [Hosting Your Own NuGet Feeds](#) ». حتی شبیه به سایت گالری NuGet رو هم میشه لوکال نصب کرد؛ [سورس باز است](#) .

نویسنده: Ara
تاریخ: ۱۳۹۲/۰۳/۱۰ ۲۱:۳۳

ممنون

وب هاست نمی‌خواستم ، در مورد نحوه ساختن Package می‌خواستم بدونم، که فکر می‌کردم دنگ فنگ زیادی داره

نمی‌دونستم براحتی دستور زیره

nuget pack ProjectFile.csproj

که پکیج هام رو ساختم و گذاشتم تو یک فولدر تو ویژوال استودیو تو قسمت

Options/Package Manger/Package Resources

اضافه اش کردم

البته این هم خوب بود برای ویرایش و غیره

<http://docs.nuget.org/docs/creating-packages/using-a-gui-to-build-packages>

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۳:۲۱

با سلام.

چرا وقتی که بر روی سولوشن کلیک راست میکنم گزینه Enable NuGet Package Restore وجود ندارد. nuget ولی نصب شده است. با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۳:۳۴

- این گزینه در آخرین نگارش‌های NuGet اضافه شده. نیاز است خود NuGet رو [به روز کنید](#) .
- همچنین اگر مطابق شکل دوم، پوشه nuget. در پروژه موجود باشد، این گزینه ظاهر نخواهد شد.

نویسنده: منیره محمدی
تاریخ: ۲۰:۵۷ ۱۳۹۲/۰۸/۲۶

سلام من وقتی این خط را اجرا میکنم Install-Package structuremap این خطا را بهم میدهد

```
PM> Install-Package structuremap
```

'Install-Package : The remote name could not be resolved: 'az320820.vo.msecnd.net

At line:1 char:16

Install-Package <<<< structuremap +

CategoryInfo : NotSpecified: (:) [Install-Package], WebException +

FullyQualifiedErrorId : NuGetCmdletUnhandledException,NuGet.PowerShell.Commands.InstallPackageCommand +

نویسنده: وحید نصیری
تاریخ: ۲۱:۱۰ ۱۳۹۲/۰۸/۲۶

remote name could not be resolved یعنی مشکل DNS و یا تنظیمات اتصالی را دارید به احتمال زیاد. آدرس زیر را در IE امتحان کنید (از این جهت که تنظیمات اینترنت IE به برنامه‌های دات نت به صورت پیش فرض اعمال می‌شوند). اگر باز نشد، مشکل اتصالی دارید:

<https://az320820.vo.msecnd.net/packages/structuremap.2.6.4.1.nupkg>

نویسنده: وحید نصیری
تاریخ: ۱۳:۱۳ ۱۳۹۲/۱۱/۱۰

یک نکته‌ی تکمیلی

مطلب جاری توسط تیم NuGet منسوخ شده اعلام گردیده و در نگارش‌های جدید آن، به صورت خودکار و بدون نیاز به هیچگونه تنظیم اضافه‌تری انجام می‌شود. [بیشتر در اینجا](#)

نویسنده: وحید نصیری
تاریخ: ۱۳:۰ ۱۳۹۴/۰۱/۰۵

یک نکته

اگر حین ارائه‌ی برنامه‌ی خود فایل exe مربوط به nuget را ارائه ندهید، پیام خطای یافت نشدن آن را در حین Build مشاهده خواهید کرد. برای رفع آن تنها کافی است فایل NuGet.targets را گشوده و دریافت خودکار nuget.exe را فعال کنید:

```
<DownloadNuGetExe Condition=" '$(DownloadNuGetExe)' == '' ">true</DownloadNuGetExe>
```

این تنظیم به صورت پیش فرض غیرفعال است.

تغییراتی در Entity framework 6 صورت گرفته و در ذیل لیستی از موارد آن آمده است. همچنین پیشتر در همین سایت نیز به آن‌ها [اشاره‌ای شده](#) که باز تولید پروایدرها برای نسخه جدید Entity framework یکی از آن‌ها می‌باشد:

Rebuilding EF Providers for EF6
Updating Applications to use EF6
EF Tools: adding EF6 support & enabling out-of-band releases
Async Query and Save
Connection Resiliency
Code-Based Configuration
Dependency Resolution
Interception/SQL logging
Custom implementations of Equals or GetHashCode on entity classes
Custom Code First Conventions
Code First Mapping to Insert/Update/Delete Stored Procedures
Configurable Migrations History Table
Multiple Contexts per Database

اکنون برای به‌روزرسانی به نسخه جدید، جهت ادامه استفاده از SqlServer Compact مواردی باید لحاظ شود که به آن‌ها اشاره خواهیم کرد و قبل از آن‌ها رعایت یک سری از پیشنیازها لازم است. برای مثال در وب کانفیگ برای استفاده از پروایدر SqlServer Compact بعنوان پروایدر پیش فرض باید قسمت مربوطه را به نحو ذیل تغییر داده باشیم:

```
<entityFramework>
  <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlCeConnectionFactory,
EntityFramework">
    <parameters>
      <parameter value="System.Data.SqlServerCe.4.0" />
    </parameters>
  </defaultConnectionFactory>
  <providers>
    <provider invariantName="System.Data.SqlServerCe.4.0"
type="System.Data.Entity.SqlServerCompact.SqlCeProviderServices, EntityFramework.SqlServerCompact" />
  </providers>
</entityFramework>
```

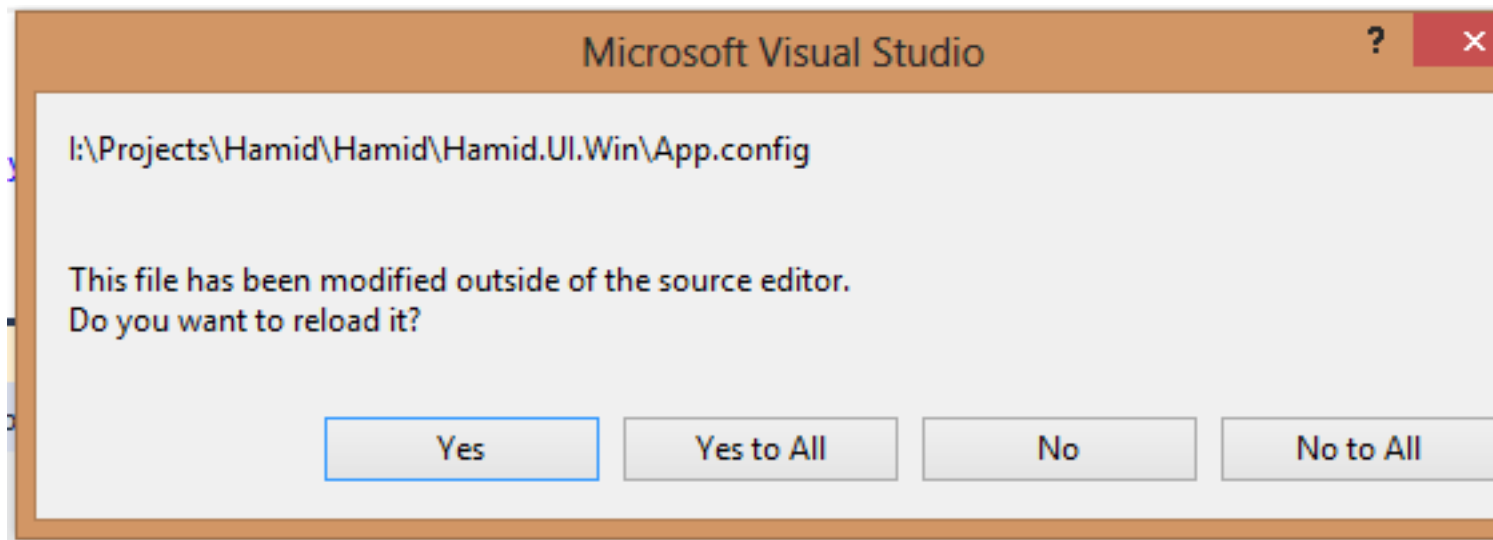
حالا در کنسول نیوگت دستور زیر را برای به‌روزرسانی فقط Entity Framework وارد و اجرا میکنیم:

```
Update-Package EntityFramework
```

پیغام موفقیت آمیز بودن به‌روز رسانی در خروجی نیوگت ظاهر می‌شود

```
PM> Update-Package EntityFramework
Updating 'EntityFramework' from version '5.0.0' to '6.0.1' in project 'Hamid.Core.Model'.
Removing 'EntityFramework 5.0.0' from Hamid.Core.Model.
Successfully removed 'EntityFramework 5.0.0' from Hamid.Core.Model.
```

و نیز تاییدی برای اعمال تغییرات به‌روز رسانی Entity framework انجام میشود تا فایل کانفیگ پروژه را تغییر دهد:



این تغییرات شامل موارد ذیل می‌باشند (در صورت به‌روز رسانی دستی، منظور کپی پکیج بصورت دستی، اعمال تغییرات در کانفیگ‌ها مورد نیاز است):

```
<!-- 1. Change in <configuration><configSections> -->
<section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=4.4.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
<section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
<!-- 2. Add in <entityFramework><providers> -->
<provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
```

بعد از به‌روز رسانی Entityframework باید پکیج EntityFramework.SqlServerCompact برای ادامه استفاده از پروایدر نصب شود که با دستور نیوگت زیر این امر نیز میسر است:

```
PM> Install-Package EntityFramework.SqlServerCompact
```

حالا بدون مشکل می‌توان از پروژه بیلد گرفت و کار توسعه را ادامه داد.

بهبود سرعت دریافت بسته‌های نیوگت

در کشور بسیاری از اوقات دسترسی به پروتکل HTTPS به کندی صورت می‌گیرد. گاهی از اوقات نیز این دسترسی غیر ممکن می‌شود تا حد دریافت چند بایت در دقیقه. همین مساله تاکنون بر روی بسیاری از مسایل دیگر نیز تاثیر گذار بوده است؛ برای مثال اگر یک مخزن کد را مثلا در CodePlex یا GitHub داشته باشید، چون تمام Commitها از طریق همین پروتکل امن صورت می‌گیرد، کار کردن با آنها بسیار مشکل خواهد شد. نمونه‌ی دیگر آن دسترسی به NuGet است. فید NuGet در VS.NET به https تنظیم شده است. اگر دسترسی به https برای شما به کندی صورت می‌گیرد فقط کافی است مسیر فید آنرا در منوی Tools، گزینه‌ی Options، ذیل قسمت Package manager یافته و به <http://nuget.org/api/v2> تغییر دهید؛ یعنی به http خالی، بجای Http s؛ تا سرعت دریافت بسته‌های NuGet مورد نظر افزایش یابند.

دریافت مستقیم بسته‌های نیوگت

برای دریافت بسته‌های نیوگت که دارای پسوند nupkg هستند، اما در اصل یک فایل zip بیشتر نیستند، الزامی به استفاده از ابزار و افزونه نیوگت در VS.NET نیست. می‌توان این بسته‌ها را به صورت مستقیم نیز دریافت کرد. برای مثال اگر آدرس بسته‌ای در سایت NuGet به صورت زیر است:

<https://www.nuget.org/packages/PropertyChanged.Fody>

برای دریافت مستقیم آن کافی است آدرس ذیل را درخواست کنید:

<https://www.nuget.org/api/v2/package/PropertyChanged.Fody/1.42.0>

یک api/v2 به این لینک اضافه می‌شود به همراه شماره نگارش مدنظر برای دریافت:

<https://www.nuget.org/api/v2/package/{packageID}/{packageVersion}>

و یا برای مثال در [سایت نیوگت](http://www.nuget.org) عضو شوید و سپس به آن لاگین کنید. به این ترتیب با مراجعه به هر کتابخانه‌ای که در آنجا آپلود شده، یک لینک download در کنار صفحه، سمت چپ ظاهر می‌شود. با کلیک بر روی آن فایل nupkg آن کتابخانه قابل دریافت خواهد بود. این فایل در حقیقت یک فایل zip است. بنابراین کار کردن با محتویات آن ساده‌است. به صورت خلاصه:

لینک اصلی کتابخانه: <https://www.nuget.org/packages/Twitter.Bootstrap.RTL.Less/3.0.0>

لینک دانلود آن: <https://www.nuget.org/api/v2/package/Twitter.Bootstrap.RTL.Less/3.0.0>

راه دیگر، ساخت دستی این آدرس است:

<https://az320820.vo.msecnd.net/packages/propertychanged.fody.1.42.0.nupkg>

که در حقیقت تشکیل شده است از:

<https://az320820.vo.msecnd.net/packages/{name}.{version}.nupkg>

اگر نام آخرین بسته ارسالی PropertyChanged.Fody 1.42.0 باشد. فقط کافی است این دو قسمت را، یعنی نام و شماره نگارش را با یک نقطه به هم متصل کنید و سپس به انتهای آن، پسوند nupkg را اضافه نمایید. این فایل در آدرس <https://az320820.vo.msecnd.net/packages> به صورت مستقیم قابل دریافت است.

اهمیت تنظیمات IE

اگر پیام قطع شدن اتصال یا مشکلات DNS را در کنسول NuGet در VS.NET دریافت می‌کنید، ابتدا سعی کنید همان روش ذکر شده

در ابتدای بحث را امتحان کنید. اگر کار نکرد احتمالا مشکل از تنظیمات IE است. برای مثال اگر بر روی تنظیمات اتصالی شما در IE یک پروکسی غیرقابل دسترسی در زمان جاری، تنظیم شده باشد، این مساله مستقیما بر روی اتصالات برنامه‌های دات نت نیز تاثیر گذار است. بنابراین ابتدا لینک `nupkg` را که ساخته‌اید یکبار با IE امتحان کنید. اگر قابل دریافت نبود یعنی تنظیمات آن به هم ریخته است و این مساله بر روی بسیاری از برنامه‌های دیگر نیز تاثیر گذار است.

نظرات خوانندگان

نویسنده: رامین علیرضایی
تاریخ: ۱۵:۹ ۱۳۹۲/۰۹/۰۸

با سلام
ممنون از مطلب خوبتون
برای TFS چه باید کرد؟ (البته bitbucket هم همین وضع رو داره)
سه روز هستش که به صورت عادی در دسترس نیست.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۷:۳۳ ۱۳۹۲/۰۹/۰۸

استفاده از <http://nuget.org/api/v2> در اینجا چون [اجباری](#) به استفاده از HTTPS نبوده (از طرف طراحان این سرویس)، میسر شده وگرنه این تغییر فایده‌ای نداشت.

نویسنده: آریا
تاریخ: ۸:۵۰ ۱۳۹۲/۰۹/۱۰

سلام
یه راه هم هست برای دوستانی که دسترسی مستقیم به سایت Nuget ندارن. من مدت‌ها از ساخت دستی آدرس استفاده میکردم که خوب درباره بسته‌هایی که وابستگی داشتن خیلی اذیت کننده بود چون فقط خود فایل رو میگیره. ولی جایگزین خوب این قضیه فایل اجرایی Nuget Command Line هستش [اینجا](#) و تمام و وابستگی‌ها رو هم خودش دانلود میکنه!

نویسنده: محسن خان
تاریخ: ۹:۲۵ ۱۳۹۲/۰۹/۱۰

ایمن از همون سایت دانلود می‌کنه. فرقی نداره. فایل‌ها در یک آدرس بیشتر نیست.

نویسنده: آریا
تاریخ: ۱۷:۲۷ ۱۳۹۲/۰۹/۱۰

شما وقتی این آدرس رو وارد کنید

<https://www.nuget.org/api/v2/package/PropertyChanged.Fody/1.42.0>

[فقط بسته‌ی مربوطه رو میگیرید و هیچ کدوم از وابستگی‌ها دانلود نمیشن! شما چون توی تنظیماتتون آدرس‌های مربوطه Nuget مثل](#)

[/https://www.nuget.org/api/v2](https://www.nuget.org/api/v2)

رو دارید خود Nuget تمام وابستگی‌ها رو هم براتون دانلود و نصب میکنه. من هم عرض کردم در صورتیکه از داخل VS دسترسی به سایت Nuget ندارید!

اگر دسترسی نداشته باشید بسته‌های پیش نیاز دریافت نخواهند شد , در نتیجه نصب صورت نمیگیره. پس باید تک تک آدرس‌ها رو بسازید و بسته‌های پیش نیاز رو دریافت کنید ولی بوسیله‌ی Nuget.exe میتونید همه‌ی بسته‌ها رو یکجا دریافت کنید. شاید حالت خیلی خاصی باشه ولی بعضی جاها که دسترسی به اینترنت برای سیستمی که VS روش نصبه محدوده و نیاز دارن تا بسته‌ها رو بصورت آفلاین نصب کنن این متود کارآیی زیادی داره.

نویسنده: محسن خان
تاریخ: ۱۹:۲۳ ۱۳۹۲/۰۹/۱۰

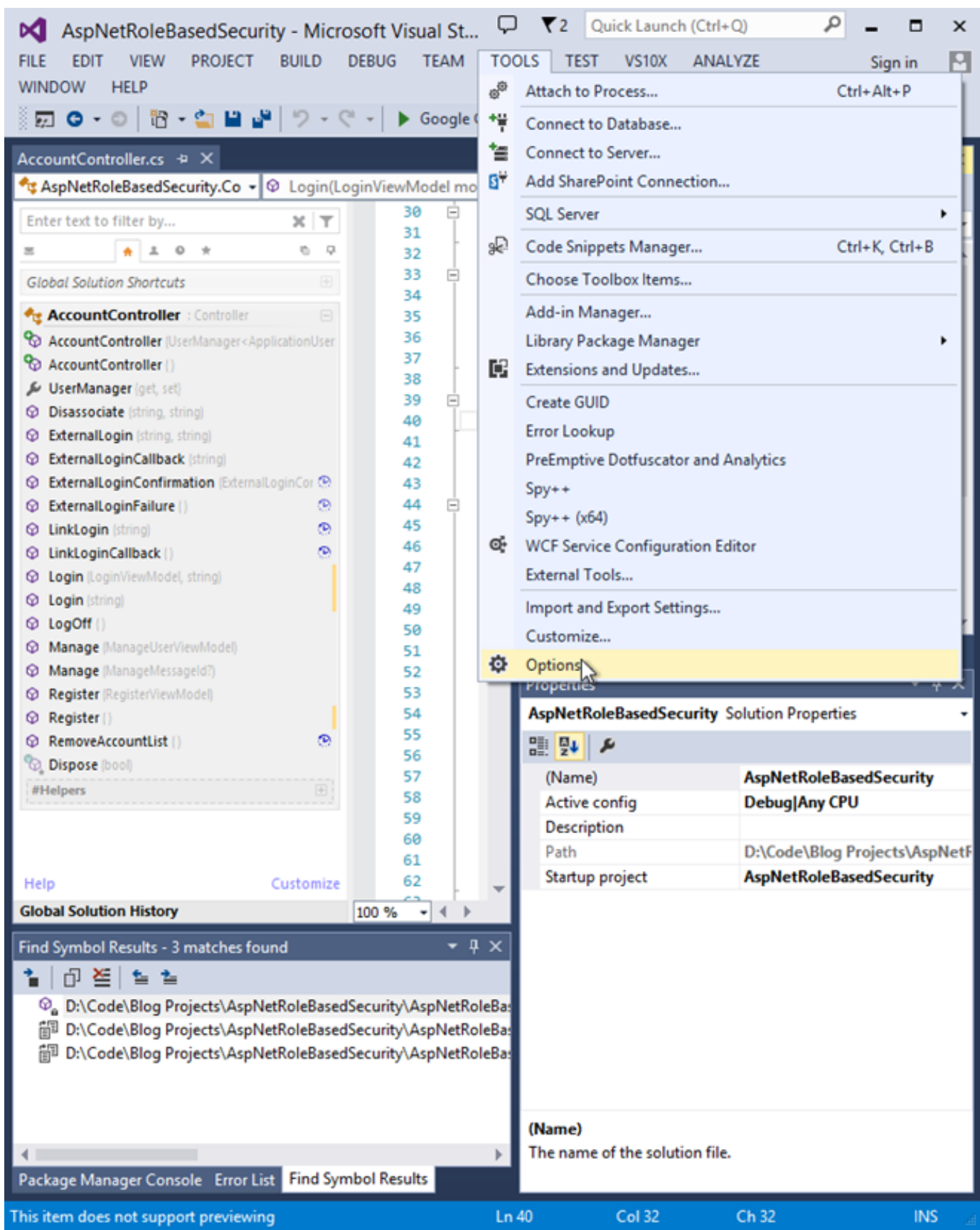
مهم نیست که وابستگی‌ها دریافت میشن یا نه. گاهی شخصی نمی‌دونه مشکلش چیه. چرا اتصال برقرار نمیشه. اصلا باید به کجا متصل بشه. الان می‌دونه که لینک نهایی‌اش کجاست و وابسته به ابزار نیست. می‌تونه خودش یک حداقل کار دیباگ رو انجام بده. می‌دونه اگر دریافتش چند بایت هست و بعد قطع میشه به چه دلیلی بوده. اگر خطای DNS دریافت می‌کنه، این به کجای سیستم ممکنه مرتبط باشه. مطلب جاری بیشتر حال و هوای دیباگ رو داره تا استفاده از ابزارهای آماده.

ابزار NuGet بسیار کار آمد و مفید است. یکی از مشکلات رایج هنگامی پیش می آید که پروژه را به همراه بسته های نصب شده به سورس کنترل push می کنید. با این کار حجم زیادی از فایل ها را به مخزن سورس کنترل آپلود می کنید و هنگام clone کردن پروژه توسط هر شخصی، این اطلاعات باید دریافت شوند. بدتر از این هنگامی است که برخی از بسته ها از سورس حذف می شوند و باید به اعضای تیم پروژه اطلاع دهید که چه بسته هایی باید دریافت و نصب شوند.

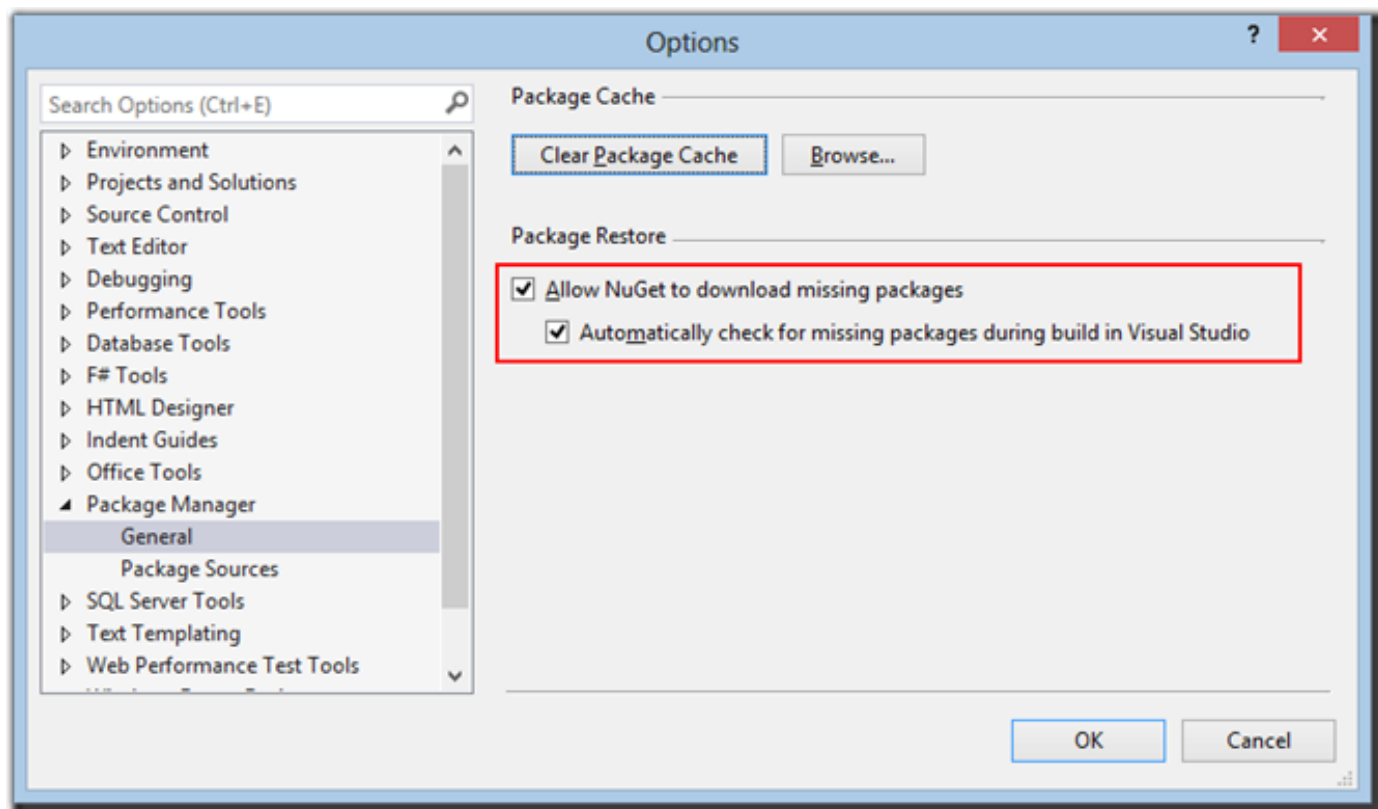
برای رفع این موارد به [NuGet Package Restore](#) وارد شوید.

به ویژوال استودیو اجازه دهید بسته های NuGet را در صورت لزوم احیا کند

پیش از آنکه بتوانیم از قابلیت [Package Restore](#) استفاده کنیم باید آن را روی ماشین خود فعال کنیم. این کار روی هر ماشین باید انجام شود (per-machine requirement). بدین منظور به منوی Package Manager -> Options -> Tools بروید.



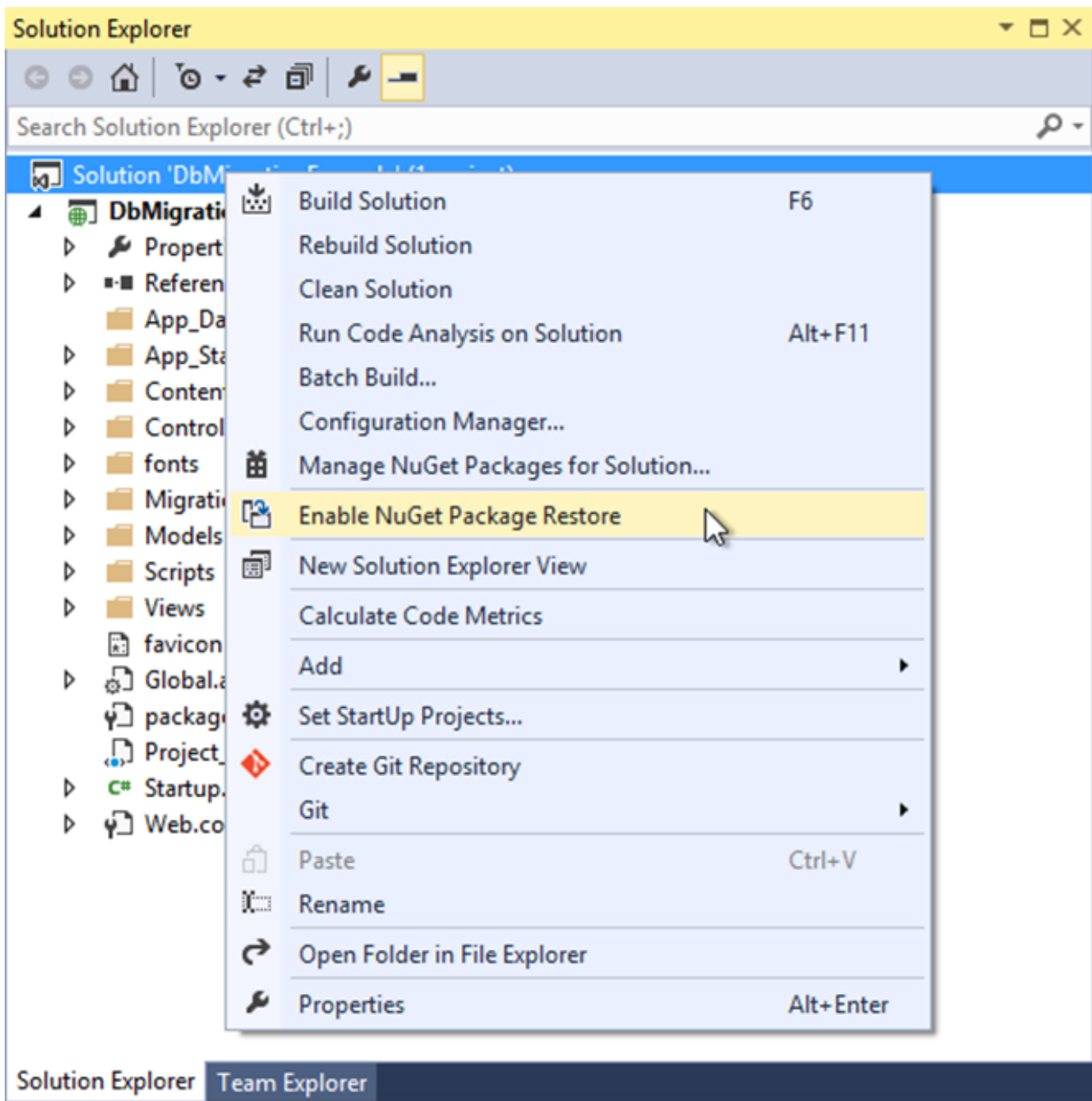
در دیالوگ باز شده تنظیمات مربوطه را مانند تصویر زیر بروز رسانی کنید.



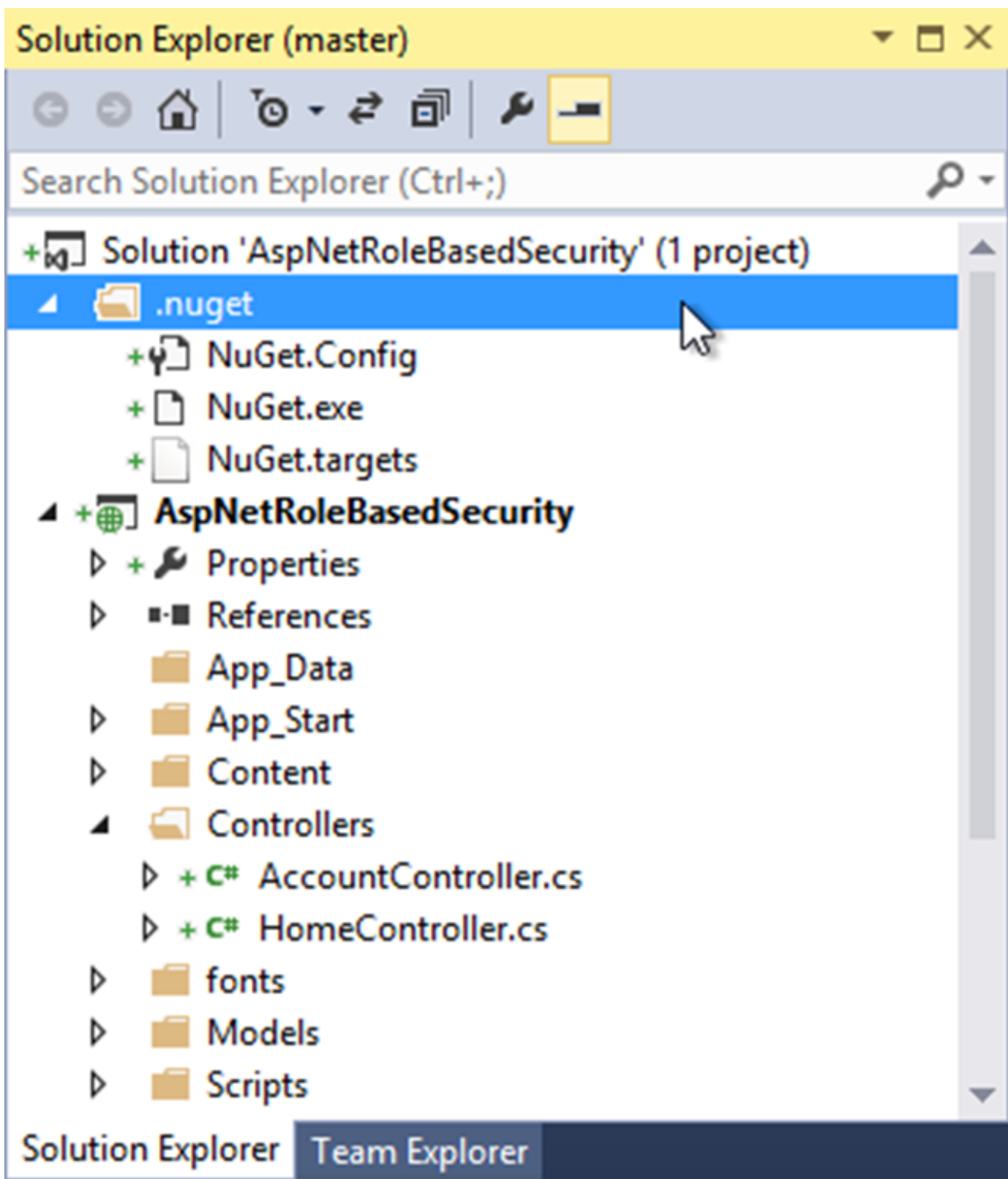
حال که ماشین ما برای بازیابی خودکار بسته‌های NuGet پیکربندی شده است، باید این قابلیت را برای Solution مورد نظر هم فعال کنیم.

فعال سازی NuGet Package Restore برای پروژه‌ها

بدین منظور روی Solution کلیک راست کنید و گزینه Enable Package Restore را انتخاب نمایید.



این کار ممکن است چند ثانیه زمان ببرد. پس از آنکه ویژوال استودیو پردازش های لازم را انجام داد، می توانید ببینید که پوشه جدیدی در مسیر ریشه پروژه ایجاد شده است.



همانطور که می بینید فایلی با نام NuGet.exe در این پوشه قرار دارد که باید به سورس کنترل آپلود شود. هنگامیکه شخصی پروژه شما را از سورس کنترل دریافت کند و بخواهد پروژه را Build کند، بسته های مورد نیاز توسط این ابزار بصورت خودکار دریافت و نصب خواهند شد.

مرحله بعد حذف کردن تمام بسته های NuGet از سورس کنترل است. برای اینکار باید فایل *gitignore* را ویرایش کنید. فرض بر

این است که سورس کنترل شما Git است، اما قواعد ذکر شده برای دیگر فریم ورک ها نیز صادق است. تنها کاری که باید انجام دهید این است که به سورس کنترل خود بگویید چه چیزهایی را در بر گیرد و از چه چیزهایی صرفنظر کند.

ویرایش فایل `.gitignore` برای حذف بسته ها و شامل کردن `NuGet.exe`

یک پروژه معمولی ASP.NET MVC 5 که توسط قالب استاندارد VS 2013 ایجاد می شود شامل 161 فایل از بسته های مختلف می شود (در زمان تالیف این پست). این مقدار قابل توجهی است که حجم زیادی از اطلاعات غیر ضروری را به مخزن سورس کنترل اضافه می کند. با استفاده از نسخه پیش فرض فایل `.gitignore` (یا فایل های مشابه دیگر برای سورس کنترل های مختلف مثل TFS) تعداد فایل هایی که در کل به مخزن سورس کنترل ارسال می شوند بیش از 200 آیتم خواهد بود. قابل ذکر است که این تعداد فایل شامل فایل های اجرایی (binary) و متعلق به ویژوال استودیو نیست. به بیان دیگر نزدیک به 75% از فایل های یک پروژه معمولی ASP.NET MVC 5 که توسط VS 2013 ساخته می شود را بسته های NuGet تشکیل می دهد، که حالا می توانند بجای ارسال شدن به مخزن سورس کنترل، بصورت خودکار بازبازی و نصب شوند.

برای حذف این فایل ها از سورس کنترل، فایل `.gitignore` را ویرایش می کنیم. اگر از سورس کنترل های دیگری استفاده می کنید نام این فایل `.hgignore` یا `.tfignore` یا غیره خواهد بود. محتوای فایل شما ممکن است با لیست زیر متفاوت باشد اما جای نگرانی نیست. تنها تغییرات اندکی بوجود خواهیم آورد و مابقی محتویات فایل مهم نیستند.

چشم پوشی از پوشه Packages

فایل `.gitignore` را باز کنید و برای نادیده گرفتن پوشه بسته های NuGet در سورس، خط زیر را به آن اضافه کنید.

```
packages*/
```

استثنای برای در نظر گرفتن `NuGet.exe` ایجاد کنید

به احتمال زیاد فایل `.gitignore` شما از فایل هایی با فرمت `exe` چشم پوشی می کند. برای اینکه بسته های NuGet بتوانند بصورت خودکار دریافت شوند باید استثنای تعریف کنیم. فایل `.gitignore` خود را باز کنید و به دنبال خط زیر بگردید.

```
*.exe
```

سپس خط زیر را بعد از آن اضافه کنید. دقت داشته باشید که ترتیب قرارگیری این دستورات مهم است.

```
*.exe  
!NuGet.exe
```

دستورات بالا به Git می گوید که فایل های `exe` را نادیده بگیرد؛ اما برای فایل `NuGet.exe` استثناء قائل شود. انجام مرحله بالا انتخابی (optional) است. اگر کسی که پروژه را از مخزن سورس کنترل دریافت می کند قابلیت Package Restore را روی Solution فعال کند ابزار `NuGet.exe` دریافت می شود. اما با انجام این مراحل دیگر نیازی به این فعالسازی نخواهد بود، پس در کار اعضای تیم هم صرفه جویی کرده اید.

اطلاع رسانی به اعضای تیم و مشتریان بالقوه

دیگر نیاز نیست بسته های NuGet را به مخزن سورس کنترل ارسال کنیم. اما باید به مخاطبین خود اطلاع دهید تا پیکربندی های لازم برای استفاده از قابلیت Package Restore را انجام دهند (مثلا در فایل `README.txt` پروژه).

نظرات خوانندگان

نویسنده: Ara

تاریخ: ۲۲:۲۷ ۱۳۹۳/۰۲/۰۷

یک کار خوب داخل دیگه اینه که یک Local Package Source در شرکت داشته باشیم که دچار گیر کردن گاه به گاه ،مشکلات nuget تو ایران که بعضی وقتها گیر می‌کنه نیافتیم و package با سرعت بالا نصب بشوند

نویسنده: آرمین ضیاء

تاریخ: ۲۳:۲ ۱۳۹۳/۰۲/۰۷

میتونه رویکرد مناسبی باشه اما بهتر است که بسته‌های مورد نیاز از سرویس‌های معتبر مثل خود NuGet.org دریافت بشن تا انتشارات جدید در دسترس باشند. اگر منظورتون رو درست فهمیده باشم با این رویکرد یک کپی محلی از بسته‌ها خواهیم داشت. در صورتی که بسته‌ها نیاز به بروز رسانی داشته باشند نهایتا باز نیاز به دریافت پکیج‌ها از اینترنت است.

نویسنده: مسعود دانش پور

تاریخ: ۱۰:۱۱ ۱۳۹۳/۰۲/۰۸

به نظر بنده اگر به تایتل این نوشته مفید به "بیرون نگاه داشتن پکیج‌های NuGet از سورس کنترل Git" تغییر کنه بسیار عالی‌تر خواهد شد.

نویسنده: آرمین ضیاء

تاریخ: ۱۷:۴۹ ۱۳۹۳/۰۲/۰۸

با تشکر، عنوان بروز رسانی شد.

روش‌های زیادی برای تهیهی یک بسته‌ی نیوگت وجود دارند، مانند استفاده از برنامه‌ی [NuGet Package Explorer](#) و یا تهیهی یک فایل nuspec و تغییر مداوم جزئیات آن، به ازای هر نگارش جدید پروژه. در ادامه قصد داریم روش خودکار سازی این تغییرات را بررسی کنیم.

الف) تهیه فایل nuspec

NuGet قابلیت پذیرش متغیرهای خود تکمیل شونده‌ای را نیز دارد. فایل nuspec یا جزئیات بسته‌ی تولیدی نیوگت، در این حالت یک چنین شکلی را پیدا می‌کند:

```
<?xml version="1.0"?>
<package >
  <metadata>
    <id>$id$</id>
    <version>$version$</version>
    <title>$title$</title>
    <authors>$author$</authors>
    <owners>$author$</owners>
    <licenseUrl>https://site.com/prj/LICENSE</licenseUrl>
    <projectUrl>https://site.com/prj</projectUrl>
    <requireLicenseAcceptance>false</requireLicenseAcceptance>
    <description>$description$</description>
    <copyright>Copyright 2015 My Name</copyright>
  </metadata>
</package>
```

برای مثال اگر پروژه‌ی ما TestNuget نام دارد، فایلی به نام TestNuget.nuspec را با محتویات فوق برای آن تهیه خواهیم کرد. همانطور که مشاهده می‌کنید، در این فایل ویژه، هیچکدام از اطلاعات شماره نگارش پروژه، نام نویسنده و غیره، از پیش تعیین نشده‌اند. این اطلاعات، از فایل AssemblyInfo.cs پروژه، به صورت خودکار دریافت خواهند شد و نکته‌ی مهم آن، قرار گرفتن این فایل nuspec در پوشه‌ی اصلی پروژه است. از این جهت که برای کامپایل آن و تبدیل به یک بسته‌ی نیوگت، فایل nuget.exe بر روی فایل پروژه‌ی اصلی برنامه اجرا خواهیم کرد و نه بر روی این فایل nuspec. بنابراین تکمیل اطلاعات فایل AssemblyInfo.cs را فراموش نکنید. برای مثال اطلاعات AssemblyCompany آن، در قسمت authors فایل فوق جایگزین می‌شود.

ب) تهیه فایل NuGet.exe

فایل NuGet.exe را جهت کامپایل فایل فرضی TestNuget.nuspec نیاز داریم. می‌توان آن را [از سایت کدپلکس](#) تهیه کرد و یا تنها کافی است بر روی Solution موجود در VS.NET کلیک راست کرده و گزینه‌ی Enable NuGet Package Restrore را انتخاب کنیم. با اینکار به صورت خودکار پوشه‌ی ویژه‌ای به نام nuget. به همراه فایل NuGet.exe ایجاد می‌شود.

ج) کامپایل فایل nuspec

در همان پوشه‌ی جدید nuget.، یک فایل bat را با محتوای ذیل تهیه کنید:

```
"%~dp0NuGet.exe" pack "..\TestNuGet\TestNuGet.csproj" -Prop Configuration=Release
copy "%~dp0*.nupkg" "%localappdata%\NuGet\Cache"
pause
```

در این مثال، مسیر TestNuGet\TestNuGet.csproj به محل قرارگیری فایل پروژه‌ی برنامه اشاره می‌کند. در اینجا فایل nuget.exe را بر روی فایل csproj برنامه اجرا خواهیم کرد. با توجه به اینکه قرار است یک بسته‌ی عمومی منتشر شود، بهتر است تنظیمات تولید بسته‌ی نیوگت را بر روی حالت release قرار داد که در این دستور، قید شده‌است. در ادامه برای اینکه امکان آزمایش محلی این بسته را نیز داشته باشیم، فایل بسته‌ی تولیدی، به کش محلی نیوگت، در سیستم جاری کپی می‌شود.

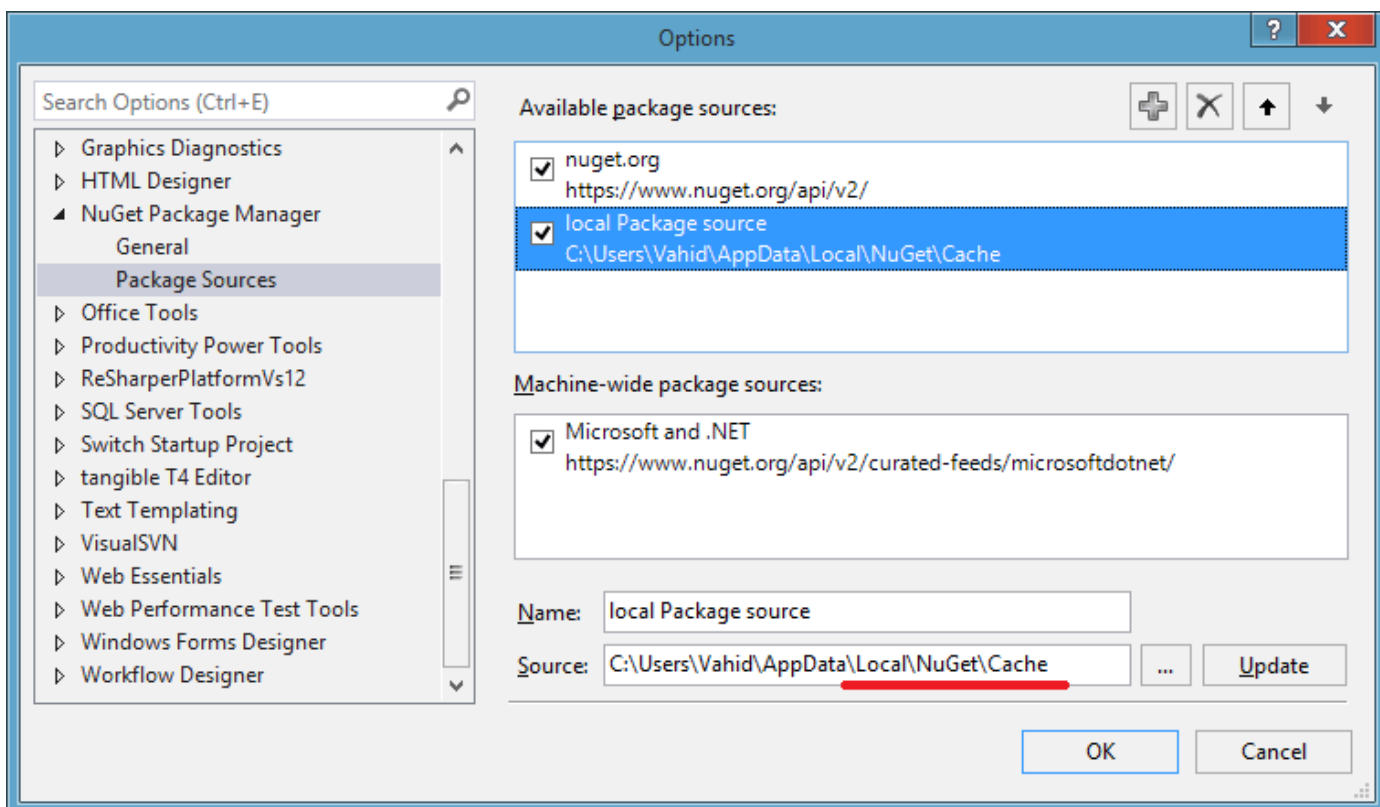
نکته‌ی جالب این روش، تهیهی قسمت dependencies پروژه، به صورت خودکار است:

```
<?xml version="1.0"?>
<package xmlns="http://schemas.microsoft.com/packaging/2011/08/nuspec.xsd">
  <metadata>
    <id>TestNuget</id>
    <version>1.0.0.0</version>
    <title>TestNuget</title>
    <authors>Vahid N.</authors>
    <owners>Vahid N.</owners>
    <licenseUrl>https://site.com/prj/LICENSE</licenseUrl>
    <projectUrl>https://site.com/prj</projectUrl>
    <requireLicenseAcceptance>false</requireLicenseAcceptance>
    <description>This is a test.</description>
    <copyright>Copyright 2015 My Name</copyright>
    <dependencies>
      <dependency id="EntityFramework" version="6.1.2" />
    </dependencies>
  </metadata>
</package>
```

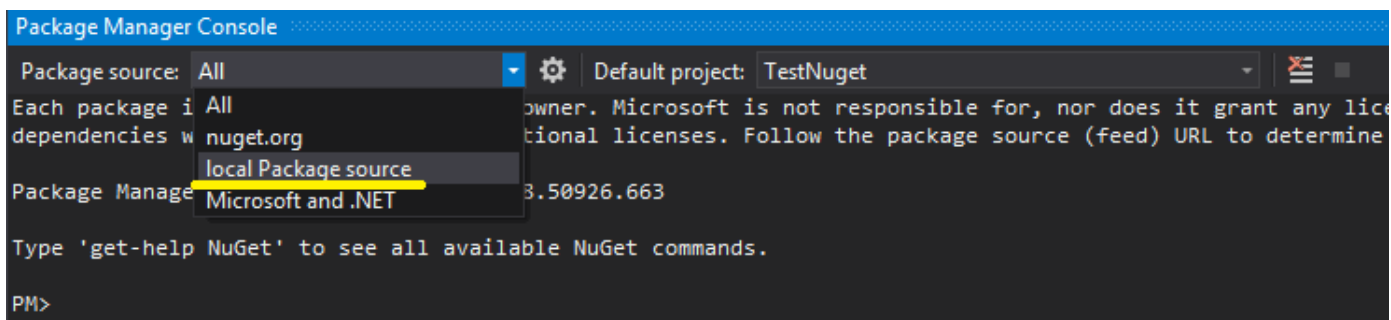
اطلاعاتی را که در اینجا مشاهده می‌کنید، حاصل کامپایل فایل nuspec معرفی شده‌ی در ابتدای بحث است (با اجرای فایل bat تهیه شده) که یک کپی از آن در فایل TestNuget.1.0.0.0.nupkg نهایی نیز قرار می‌گیرد. به این ترتیب دیگر نیازی نیست تا این قسمت را به صورت دستی معرفی و هر بار با به روز رسانی وابستگی‌های پروژه، شماره نگارش‌های آن‌ها را نیز به روز کرد. اطلاعات version به صورت خودکار از پروژه‌ی جاری استخراج می‌شوند.

د) آزمایش محلی بسته‌ی جدید

اکنون که فایل TestNuget.1.0.0.0.nupkg تولیدی را در آدرس %localappdata%\NuGet\Cache نیز کپی کرده‌ایم، به منوی Tools/Options مراجعه و در قسمت NuGet Package Manager آن، به قسمت Package sources، این آدرس کش محلی را نیز معرفی کنید:



به این ترتیب، بدون ارسال یک بسته به سایت اصلی نیوگت، می‌توان بسته را از کش محلی نیوگت نصب و آزمایش کرد:



فایل‌های ذکر شده در مطلب فوق را از اینجا می‌توانید دریافت کنید:

[TestNuget.zip](#)

نظرات خوانندگان

نویسنده: آرش کرمی
تاریخ: ۱۹:۲۳ ۱۳۹۳/۱۲/۰۵

اگر بخواهیم به صورت اتوماتیک بعد از هر Build یا Check-in این کار انجام بپذیرد چه باید کرد؟
و اینکه آیا پس از آپدیت شدن nupackها در صورتی که در یک سولوشن Enable nuget packet restore داشته باشیم آیا بعد از Build آن سولوشن، پکیج هایش نیز آپدیت میشود؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۹:۳۲ ۱۳۹۳/۱۲/۰۵

- در [post build event](#) می‌توان فایل bat ایی را که در متن به آن اشاره شده، به صورت خودکار اجرا کرد؛ یک مثال عمومی:
`call "$(ProjectDir)batch\run.bat"`

- خیر. کار [Enable nuget packet restore](#) فقط رفع نواقص بسته‌های مرتبط با پروژه‌ی جاری است. برای به روز رسانی بسته‌های موجود باید دستور PM>Update-Package جداگانه صادر شود (هربار که نیاز است).

زمانیکه پروژه‌ی شما وابستگی‌های متعددی داشته باشد، احتمال برخوردن به یک چنین خطایی بسیار محتمل است:

Could not load file or assembly Newtonsoft.Json or one of its dependencies. The system cannot find the file specified.

کتابخانه‌ی Newtonsoft.Json جزو پروژه‌هایی است که مدام به روز رسانی و نگهداری می‌شود. در این بین ممکن است وابستگی A از نگارش 4.5 آن استفاده کند و وابستگی B بر اساس نگارش 4.7 آن کامپایل شده باشد و وابستگی جدیدی از نگارش 6 آن استفاده کند. در یک چنین حالتی زمانیکه پروژه‌ی شما آخرین نگارش JSON.NET را به همراه داشته باشد، وابستگی‌های A و B پیام یافت نشدن نگارش‌های خاص کتابخانه‌ی Newtonsoft.Json را صادر خواهند کرد و در این حالت، برنامه در همان ابتدای کار کرش می‌کند. برای رفع این مشکل می‌توان به دات نت گفت که تمام درخواست‌های مرتبط با JSON.NET را به اسمبلی خاصی هدایت و پردازش کن. به این قابلیت assembly redirects گفته می‌شود و به نحو ذیل در فایل‌های کانفیگ برنامه (app.config یا web.config) قابل تعریف است:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <!-- ... -->
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

به این ترتیب تمام اسمبلی‌هایی که بر اساس نگارش قدیمی JSON.NET کامپایل شده‌اند، باور خواهند کرد که نگارش 6 آن، همان نگارشی است که باید با آن کار کنند.

به روز رسانی قسمت assembly redirects توسط NuGet

البته اگر بسته‌ی جدیدی را توسط نیوگت نصب کنیم، این assembly redirects به صورت خودکار توسط آن اضافه خواهند شد اما ... گاهی ممکن است موارد قدیمی را به روز نکند یا موردی فراموش شوند یا حتی اگر اطلاعاتی را پیشتر از وب یافته‌اید، دارای خطای تایپی باشند. برای یک چنین مواردی اگر با خطای Could not load file or assembly ابتدای بحث مواجه شدید، مراحل ذیل را طی کنید تا بتوانید قسمت assembly redirects را بازسازی کنید:

الف) به فایل config برنامه مراجعه کرده و کلا قسمت assemblyBinding آن را حذف کنید.

ب) سپس دستور ذیل را در کنسول پاورشل نیوگت اجرا کنید:

```
PM> Get-Project -All | Add-BindingRedirect
```

این دستور به تمام پروژه‌های موجود در solution جاری مراجعه کرده و قسمت assemblyBinding آن‌ها را بر اساس آخرین اطلاعات پروژه‌ها و وابستگی‌های آن‌ها به روز می‌کند.