

در C# می‌توانید در انتهای تعریف آخرین آیتم یک Enum یا هنگام استفاده از سینتکس Object Initializer یا Collection Initializer، یک کامای اضافی قرار بدید.

اون طور که [گفته شده](#)، این رفتار بدین دلیل است که Code Generator ها راحت‌تر بتوانند کد تولید کنند. مطمئناً اگر در یک حلقه‌ی تکرار برای ایجاد آیتم‌های یک Enum، در انتهای آیتم‌های اون، کاراکتر ";" قرار می‌دید، حذف نکردن آخرین کاما از حذف کردن اون کار راحت‌تری است! همچنین Comment کردن آخرین آیتم نیز راحت‌تر صورت می‌پذیرد.

```
public enum MyEnum
{
    Item1 = 1,
    Item2 = 2,
    Item3 = 4,
    // Item4
}

MyViewModel viewModel = new MyViewModel()
{
    Property1 = "Value1",
    Property2 = "Value2",
    Property3 = "Value3",
};
```

و البته، در هنگام فراخوانی یک متد میشه به تعداد دلخواه در انتهای اون، کاراکتر ";" قرار داد.

```
myBusiness business = new myBusiness();
business.DoWork(); ; ; ; ; ; ;
```

نظرات خوانندگان

نویسنده:

پژمان

تاریخ:

۱۸:۲۲ ۱۳۹۱/۰۵/۱۲

این رو میشه اسمش رو گذاشت forgiveness in language design

Visual Studio Export و Import کردن Breakpoint ها

عنوان:

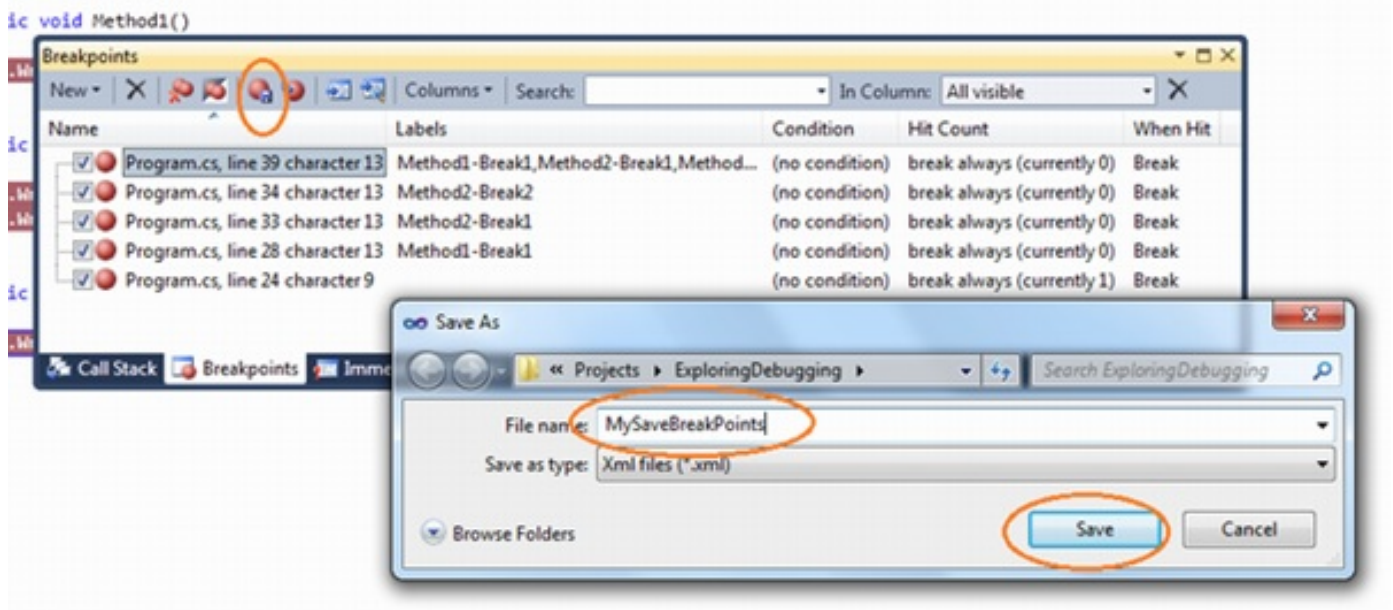
نویسنده: سیروان عقیقی

تاریخ: ۱۳۹۱/۰۹/۲۲ ۸:۳۰

آدرس: www.dotnettips.info

برچسب‌ها: Tips, Tricks, Visual Studio

ویژوال استدیو Breakpoint ها را در یک فایل XML ذخیره میکند. برای ذخیره Breakpoint ها فقط کافی است بر روی دکمه Export در پنجره Breakpoint که در شکل زیر نمایش داده شده است کلیک کنید.



شما می‌توانید فایل XML ذخیره شده را بعداً استفاده کنید و یا می‌توانید آن را به برنامه نویسان دیگر هم بدهید. اجازه دهید نگاهی داشته باشیم بر محتویات داخل فایل XML. فایل XML کلکسیونی از تگ BreakPoints داخل BreakpointCollection است. هر تگ Breakpoint حاوی اطلاعاتی در مورد یک Breakpoint خاص است.

```

MySaveBreakPoints.xml* X
<?xml version="1.0" encoding="utf-8"?>
<BreakpointCollection>
  <Breakpoints>
    <Breakpoint>
      <Version>15</Version>
      <IsEnabled>1</IsEnabled>
      <IsVisible>1</IsVisible>
      <IsEmulated>0</IsEmulated>
      <IsCondition>0</IsCondition>
      <ConditionType>WhenTrue</ConditionType>
      <LocationType>SourceLocation</LocationType>
      <TextPosition>
        <Version>4</Version>
        <FileName>.\ExploringDebugging\Program.cs</FileName>
        <startLine>38</startLine>
        <StartColumn>12</StartColumn>
        <EndLine>38</EndLine>
        <EndColumn>56</EndColumn>
        <MarkerId>0</MarkerId>
        <IsLineBased>0</IsLineBased>
        <IsDocumentPathNotFound>0</IsDocumentPathNotFound>
        <ShouldUpdateTextSpan>1</ShouldUpdateTextSpan>
        <Checksum>...</Checksum>
      </TextPosition>
    </Breakpoint>
    <Breakpoint>...</Breakpoint>
    <Breakpoint>...</Breakpoint>
    <Breakpoint>...</Breakpoint>
    <Breakpoint>...</Breakpoint>
  </Breakpoints>
</BreakpointCollection>

```

اگر شما هر زمانی همه Breakpoint ها را از کدتان حذف کردید به راحتی می‌توانید آن را تنها با کلیک بر روی Import وارد کدتان بکنید و تمام Breakpoint های ذخیره شده را بازآوری کنید.

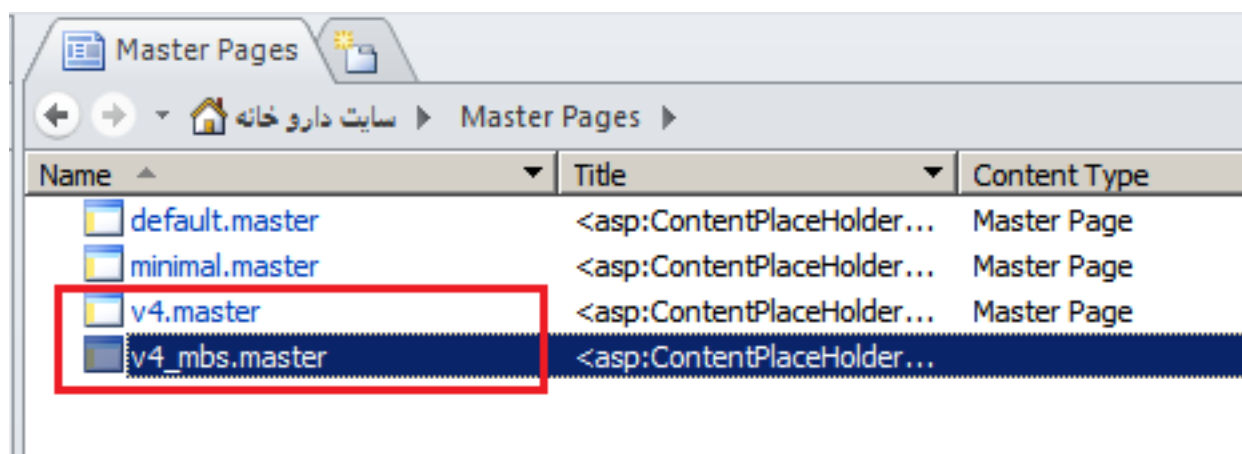
نکته: Import کردن Breakpoint براساس شماره خط کد شما می‌باشد یعنی همان خطی که شما Breakpoint را گذاشته اید پس اگر شماره خط کد شما تغییر کند Breakpoint بروی خط قبلی گذاشته میشود.

یکی از نیازهای مشتریان هنگام استفاده از سایت‌های تحت شیرپوینت ، عدم نمایش نوار مدیریتی بالای صفحه یا همان Ribbon برای کاربران ناشناس است .

شاید بتوان گفت که مزیت این راهکار نسبت به دیگر راه کارها ، تصحیح نمایش Scroll Bar مرورگر است که ممکن است در برخی روش‌ها با مشکل مواجه شود. دلیل این امر هم این است که شیرپوینت برای افزودن Ribbon به بالای صفحه Vertical Scroll Bar را از صفحه حذف می‌کند و سپس Scroll Bar سفارشی خود را به صفحه طوری اضافه می‌کند تا نوار Ribbon همیشه در بالاترین نقطه از صفحه بماند

همچنین در این روش از بارگذاری نوار هنگام بالا آمدن سایت نیز البته برای کاربران ناشناس جلوگیری می‌شود و اما روش :

سایت خود را با SharePoint Designer باز کرده و از Master Page یک کپی تهیه کنید و آن را به عنوان پیش فرض سایت تعیین کنید



سپس تگ زیر را در صفحه بیابید :

```
<div id="s4-ribbonrow">
```

و تگ زیر را به ابتدای آن (قبل از تگ) اضافه کنید :

```
<Sharepoint:SPSecurityTrimmedControl runat="server" Permissions="AddDelPrivateWebParts">
```

مانند تصویر زیر :

```
<SharePoint:DelegateControl runat="server" ControlId="GlobalNavigation"/>
<!-- Start -->
<SharePoint:SPSecurityTrimmedControl runat="server" Permissions="AddDelPrivateWebParts">

<div id="s4-ribbonrow" class="s4-pr s4-ribbonrowhidetitle">
  <div id="s4-ribboncont">
    <SharePoint:SPRibbon
      runat="server"
      PlaceholderElementId="RibbonContainer"
      CssFile="">
    <SharePoint:SPRibbonPeripheralContent
```

و تگ پایانی آن :

```
    <ContentTemplate>
      <WebPartPages:WebPartAdder ID="WebPartAdder" runat="server" />
    </ContentTemplate>
    <Triggers>
      <asp:PostBackTrigger ControlID="WebPartAdder" />
    </Triggers>
  </asp:UpdatePanel>
</div>
</div>
<!-- END -->

</SharePoint:SPSecurityTrimmedControl>

<div id="s4-workspace">
```

نکته مهم در استفاده از این تگ ، ویژگی Permissions آن است که باید با دقت و بسته به نیاز شما تعریف شود :

برخی از این موارد عبارتند از :

.EmptyMask – Has no permissions on the Web site. Not available through the user interface

.ViewListItems – View items in lists, documents in document libraries, and view Web discussion comments

.AddListItems – Add items to lists, add documents to document libraries, and add Web discussion comments

EditListItems – Edit items in lists, edit documents in document libraries, edit Web discussion comments in documents, and customize Web Part Pages in document libraries

DeleteListItems – Delete items from a list, documents from a document library, and Web discussion comments in documents

.ApproveItems – Approve a minor version of a list item or document

.OpenItems – View the source of documents with server-side file handlers

.ViewVersions – View past versions of a list item or document

.DeleteVersions – Delete past versions of a list item or document

.CancelCheckout – Discard or check in a document which is checked out to another user

.ManagePersonalViews – Create, change, and delete personal views of lists

.ManageLists – Create and delete lists, add or remove columns in a list, and add or remove public views of a list

.ViewFormPages – View forms, views, and application pages, and enumerate lists

.Open – Allow users to open a Web site, list, or folder to access items inside that container

.ViewPages – View pages in a Web site

AddAndCustomizePages – Add, change, or delete HTML pages or Web Part Pages, and edit the Web site using a SharePoint Foundation-compatible editor

.ApplyThemeAndBorder – Apply a theme or borders to the entire Web site

.ApplyStyleSheets – Apply a style sheet (.css file) to the Web site

.ViewUsageData – View reports on Web site usage

.CreateSSCSite – Create a Web site using Self-Service Site Creation

.ManageSubwebs – Create subsites such as team sites, Meeting Workspace sites, and Document Workspace sites

.CreateGroups – Create a group of users that can be used anywhere within the site collection

ManagePermissions – Create and change permission levels on the Web site and assign permissions to users and groups

BrowseDirectories – Enumerate files and folders in a Web site using Microsoft Office SharePoint Designer 2007 and WebDAV interfaces

.BrowseUserInfo – View information about users of the Web site

.AddDelPrivateWebParts – Add or remove personal Web Parts on a Web Part Page

.UpdatePersonalWebParts – Update Web Parts to display personalized information

ManageWeb – Grant the ability to perform all administration tasks for the Web site as well as manage content. Activate, deactivate, or edit properties of Web site scoped Features through the object model or through the user interface (UI). When granted on the root Web site of a site collection, activate, deactivate, or edit properties of site collection scoped Features through the object model. To browse to the Site Collection Features page and activate or deactivate site collection scoped Features through the UI, you must be a site collection administrator.

UseClientIntegration – Use features that launch client applications; otherwise, users must work on documents locally and upload changes.

UseRemoteAPIs – Use SOAP, WebDAV, or Microsoft Office SharePoint Designer 2007 interfaces to access the Web site.

ManageAlerts – Manage alerts for all users of the Web site.

CreateAlerts – Create e-mail alerts.

EditMyUserInfo – Allows a user to change his or her user information, such as adding a picture.

EnumeratePermissions – Enumerate permissions on the Web site, list, folder, document, or list item.

FullMask – Has all permissions on the Web site. Not available through the user interface.

حال خارج از تگ‌های SPSecurityTrimmedControl در ابتدا یا انتها ، باید تگ login را مانند زیر به آن اضافه کرد .

```
<!-- END -->
```

```
</SharePoint:SPSecurityTrimmedControl>
```

```
<asp:LoginView id="LoginView" runat="server">
  <AnonymousTemplate>
    <style type="text/css">
      body { overflow-y: scroll !important; overflow-x: hidden; }
      body #s4-workspace { overflow-x: hidden; overflow-y: auto !important; }
    </style>
    <!--[if lte IE 7]>
    <style type="text/css">
      html { overflow: auto !important; overflow-x: hidden; }
      body { overflow: auto !important; }
    </style>
    <![endif]-->
  </AnonymousTemplate>
</asp:LoginView>
```


و تمام :



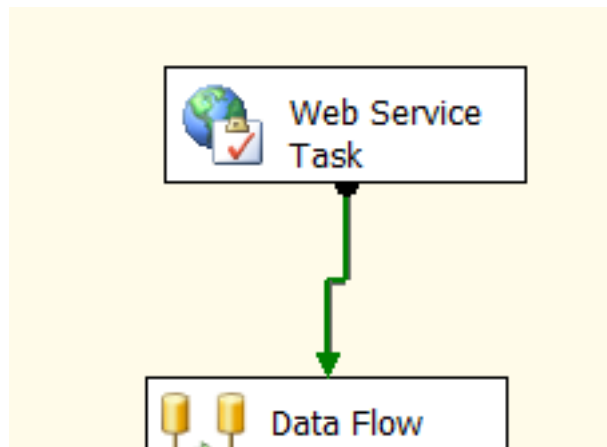
[موفق باشید](#)

عنوان: ارسال ایمیل توسط PowerShell
نویسنده: بهمن آبادی
تاریخ: ۲:۲۰ ۱۳۹۲/۰۲/۱۳
آدرس: www.dotnettips.info
برچسب‌ها: Tricks, PowerShell, Send mail, gmail

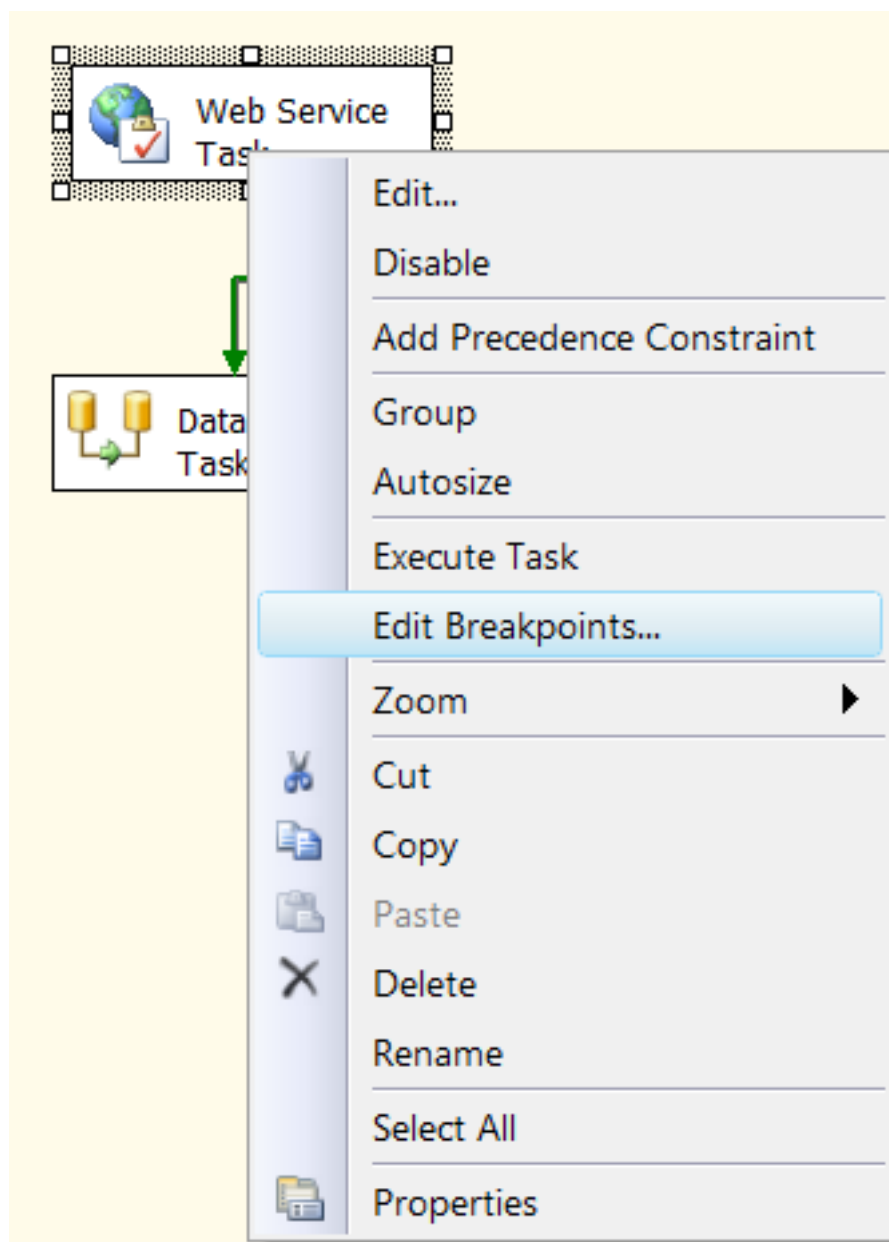
امروز می‌خواهم یکی از کاربردهای جالب PowerShell رو براتون بگم. PowerShell و اجراش کنید.
- کد زیر را ابتدا ویرایش کرده و نام کاربری و رمز عبور خود را در آن وارد نمایید. (من در اینجا از سیستم Gmail برای ارسال ایمیل استفاده کرده‌ام)
- متن دلخواه خود را در بخش \$body قرار دهید.
- در نهایت متن خود را کپی کرده و با کلیک راست بر روی PowerShell متن کپی شده در آنجا قرار می‌گیرد.
- حالا کلید Enter را فشار دهید.
ایمیل در صورتی که اطلاعات درست وارد شده باشد به ایمیل مقصد ارسال گردیده است. (:

```
$EmailFrom = "test@gmail.com"
$EmailTo = "info@dotnettips.info"
$Subject = "The subject of your email"
$Body = "What do you want your email to say"
$SMTPServer = "smtp.gmail.com"
$SMTPClient = New-Object Net.Mail.SmtpClient($SmtpServer, 587)
$SMTPClient.EnableSsl = $true
$SMTPClient.Credentials = New-Object System.Net.NetworkCredential("test", "123456");
$SMTPClient.Send($EmailFrom, $EmailTo, $Subject, $Body)
```

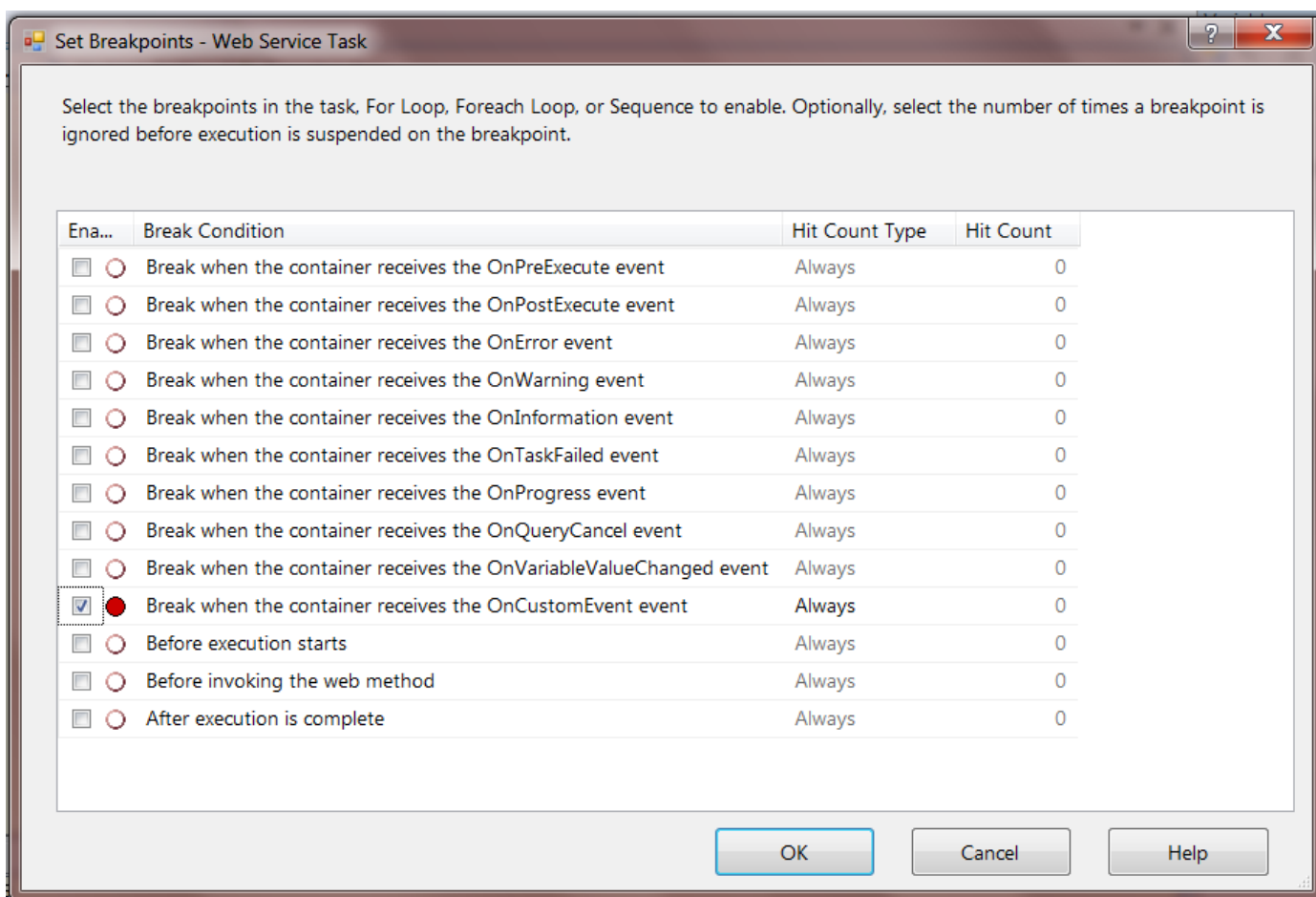
فرض کنید می‌خواهیم اطلاعات یک وب سرویس را داخل یک متغیر در package ریخته و پس از مقدار دهی ، مقدار آن متغیر آن را مشاهده کنیم . (برای اطلاع از کار با وب سرویس [به اینجا](#) مراجعه کنید) .



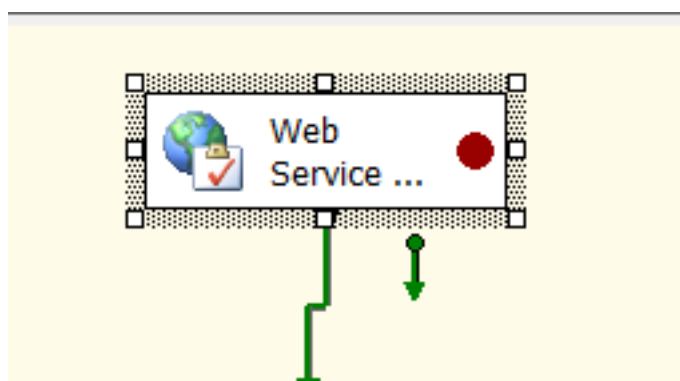
برای این کار روی کنترل کلیک سمت راست کرده و گزینه Edit Breakpoints را انتخاب می‌کنیم :



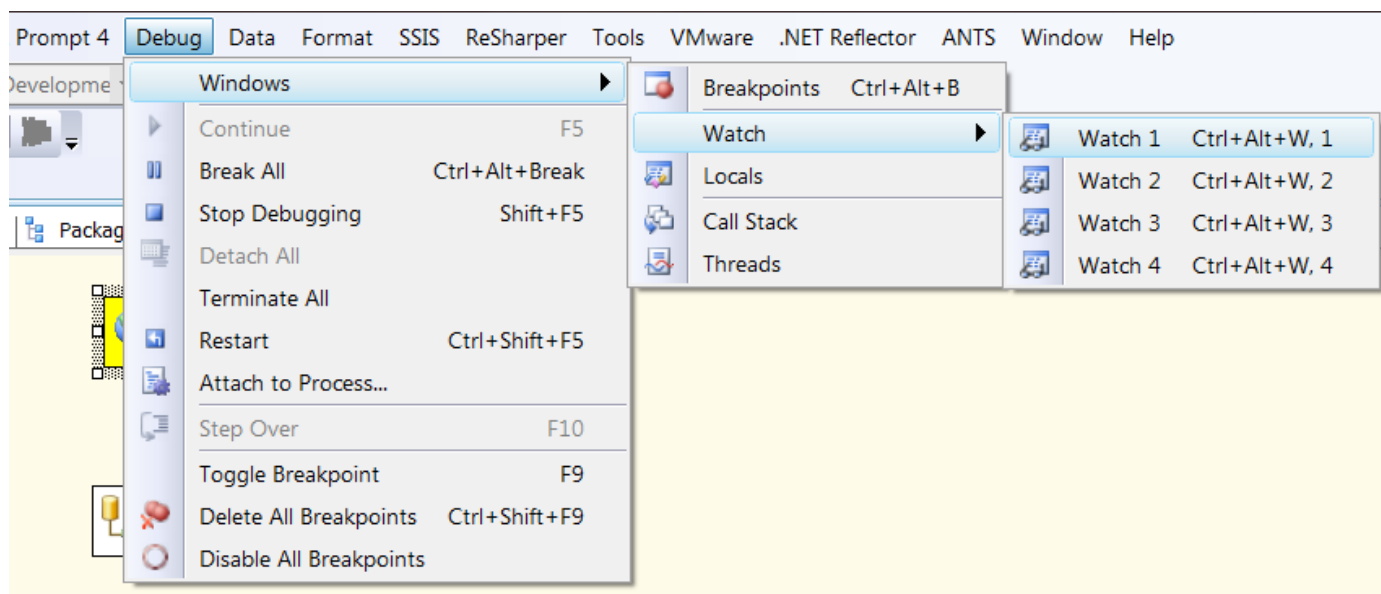
در پنجره Set Breakpoints گزینه هایی که می خواهیم در آنها break انجام شود را انتخاب می کنیم . لازم به ذکر است که این موارد بسته به کنترل های مختلف تا حدود کمی با هم فرق می کنند



پس از انتخاب گزینه یا گزینه‌های مورد نظر و بستن پنجره یک آیکون کنار کنترل نمایش داده می‌شود .



اکنون می‌توانید با باز کردن پنجره watch پس از آغاز شدن debug به مشاهده مقادیر مورد نظر بپردازید . در پنجره watch می‌توانید نام متغیر را وارد کنید تا برای شما نمایش داده شود.



| Watch 1 | | |
|---------|-----------|-------|
| | Name | Value |
| + | User::str | {C } |
| | | |