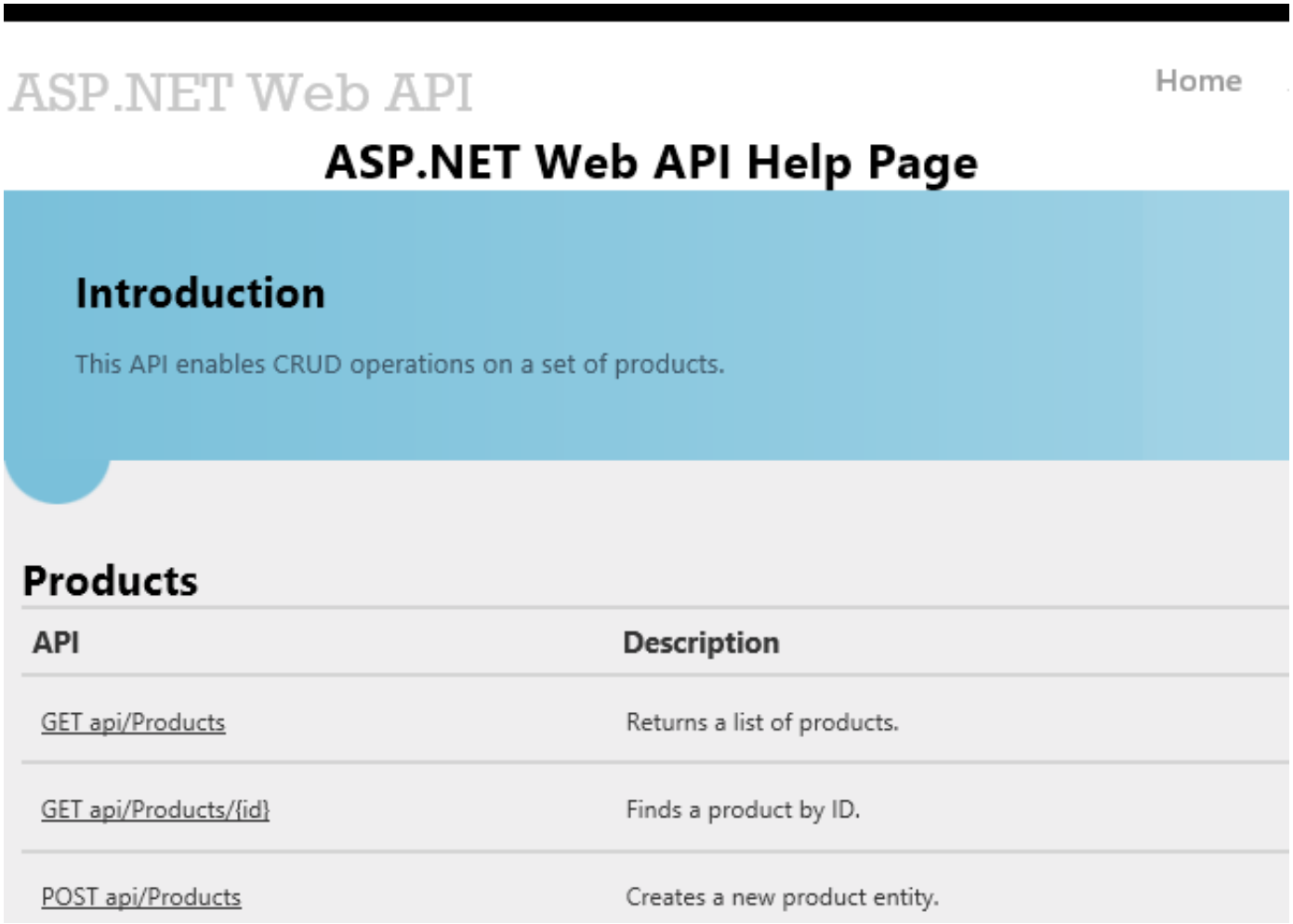
عنوان: **ایجاد صفحات راهنما برای ASP.NET Web API**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۱/۰۳ ۰:۵۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, ASP.NET Web API, Web API

وقتی یک Web API می‌سازید بهتر است صفحات راهنمایی هم برای آن در نظر بگیرید، تا توسعه دهندگان بدانند چگونه باید سرویس شما را فراخوانی و استفاده کنند. گرچه می‌توانید مستندات را بصورت دستی ایجاد کنید، اما بهتر است تا جایی که ممکن است آنها را بصورت خودکار تولید نمایید.

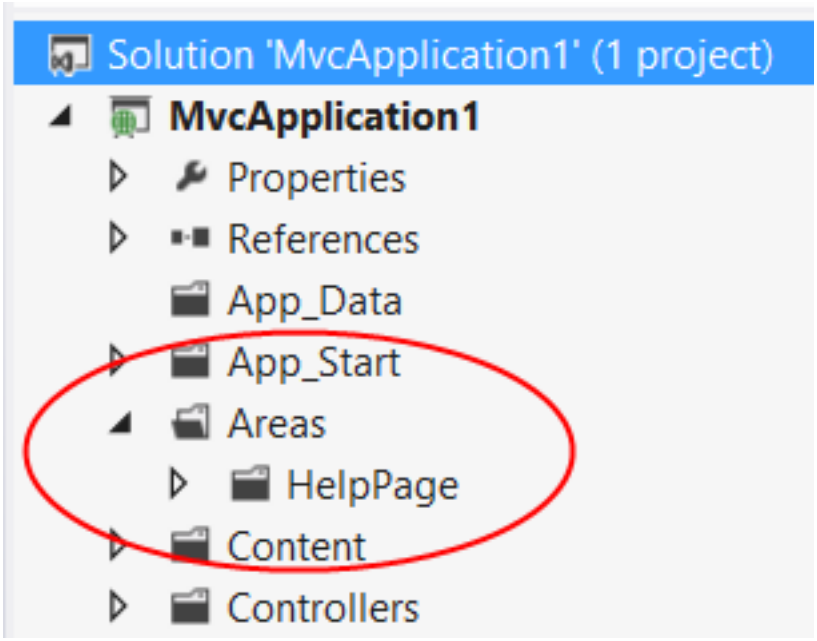
بدین منظور فریم ورک ASP.NET Web API کتابخانه ای برای تولید خودکار صفحات راهنما در زمان اجرا (run-time) فراهم کرده است.



ایجاد صفحات راهنمای API

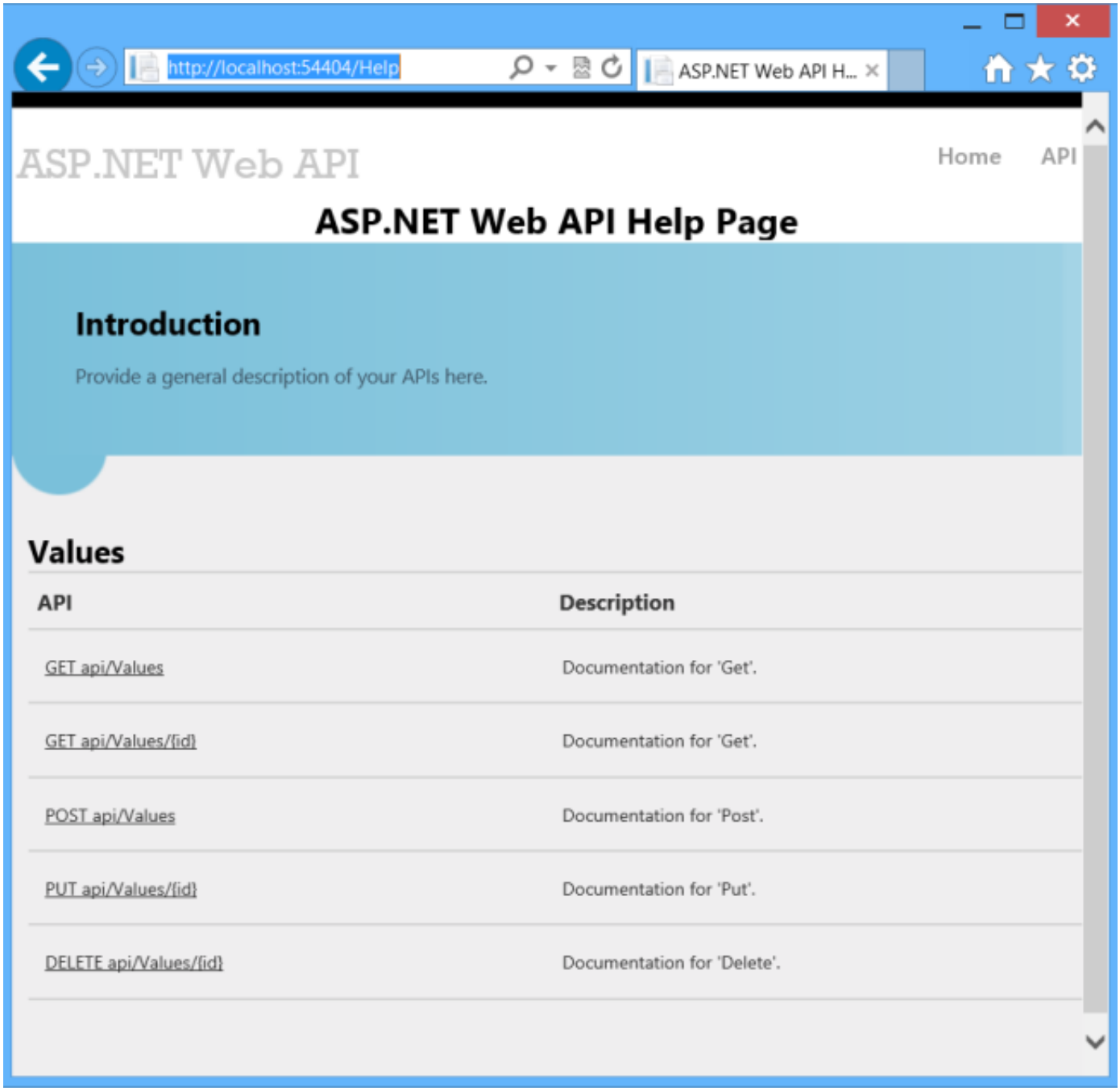
برای شروع ابتدا ابزار ASP.NET and Web Tools 2012.2 Update را نصب کنید. اگر از ویژوال استودیو 2013 استفاده می‌کنید این ابزار بصورت خودکار نصب شده است. این ابزار صفحات راهنما را به قالب پروژه‌های ASP.NET Web API اضافه می‌کند.

یک پروژه جدید از نوع ASP.NET MVC Application بسازید و قالب Web API را برای آن انتخاب کنید. این قالب پروژه کنترلری بنام ValuesController را بصورت خودکار برای شما ایجاد می‌کند. همچنین صفحات راهنمای API هم برای شما ساخته می‌شوند. تمام کد مربوط به صفحات راهنما در قسمت Areas قرار دارند.



اگر اپلیکیشن را اجرا کنید خواهید دید که صفحه اصلی لینکی به صفحه راهنمای API دارد. از صفحه اصلی، مسیر تقریبی /Help خواهد بود.

این لینک شما را به یک صفحه خلاصه (summary) هدایت می‌کند.



نمای این صفحه در مسیر Areas/HelpPage/Views/Help/Index.cshtml قرار دارد. می‌توانید این نما را ویرایش کنید و مثلا قالب، عنوان، استایل‌ها و دیگر موارد را تغییر دهید.

بخش اصلی این صفحه متشکل از جدولی است که APIها را بر اساس کنترلر طبقه بندی می‌کند. مقادیر این جدول بصورت خودکار و توسط اینترفیس **IApiExplorer** تولید می‌شوند. در ادامه مقاله بیشتر درباره این اینترفیس صحبت خواهیم کرد. اگر کنترلر جدیدی به API خود اضافه کنید، این جدول بصورت خودکار در زمان اجرا بروز رسانی خواهد شد.

ستون "API" متد HTTP و آدرس نسبی را لیست می‌کند. ستون "Documentation" مستندات هر API را نمایش می‌دهد. مقادیر این ستون در ابتدا تنها placeholder-text است. در ادامه مقاله خواهید دید چگونه می‌توان از توضیحات XML برای تولید مستندات استفاده کرد.

هر API لینکی به یک صفحه جزئیات دارد، که در آن اطلاعات بیشتری درباره آن قابل مشاهده است. معمولا مثالی از بدنه‌های درخواست و پاسخ هم ارائه می‌شود.

افزودن صفحات راهنما به پروژه ای قدیمی

می توانید با استفاده از NuGet Package Manager صفحات راهنمای خود را به پروژه‌های قدیمی هم اضافه کنید. این گزینه مخصوصا هنگامی مفید است که با پروژه ای کار می‌کنید که قالب آن Web API نیست.

از منوی Tools گزینه‌های Package Manager Console, Library Package Manager را انتخاب کنید. در پنجره Package Manager Console فرمان زیر را وارد کنید.

```
Install-Package Microsoft.AspNet.WebApi.HelpPage
```

این پکیج اسمبلی‌های لازم برای صفحات راهنما را به پروژه اضافه می‌کند و نماهای MVC را در مسیر Areas/HelpPage می‌سازد.

اضافه کردن لینکی به صفحات راهنما باید بصورت دستی انجام شود. برای اضافه کردن این لینک به یک نمای Razor از کدی مانند لیست زیر استفاده کنید.

```
@Html.ActionLink("API", "Index", "Help", new { area = "" }, null)
```

همانطور که مشاهده می‌کنید مسیر نسبی صفحات راهنما "/Help" می‌باشد. همچنین اطمینان حاصل کنید که ناحیه‌ها (Areas) بدرستی رجیستر می‌شوند. فایل Global.asax را باز کنید و کد زیر را در صورتی که وجود ندارد اضافه کنید.

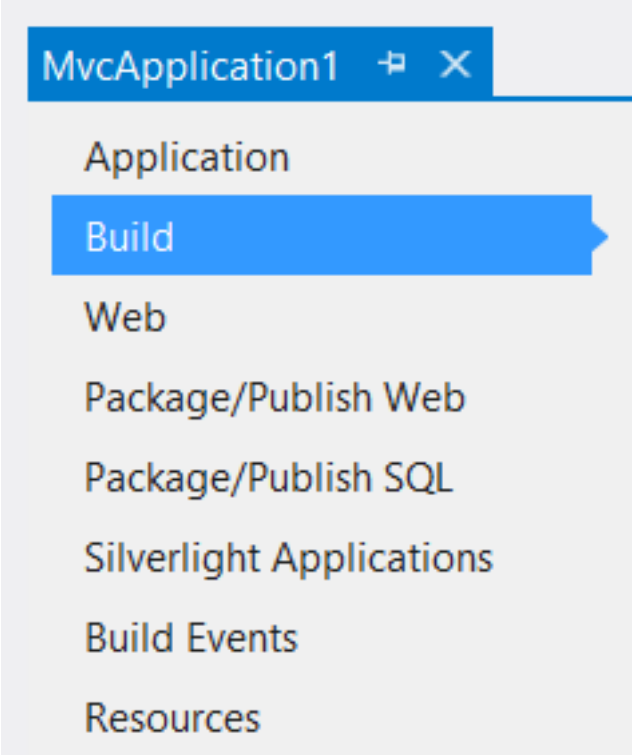
```
protected void Application_Start()
{
    // Add this code, if not present.
    AreaRegistration.RegisterAllAreas();

    // ...
}
```

افزودن مستندات API

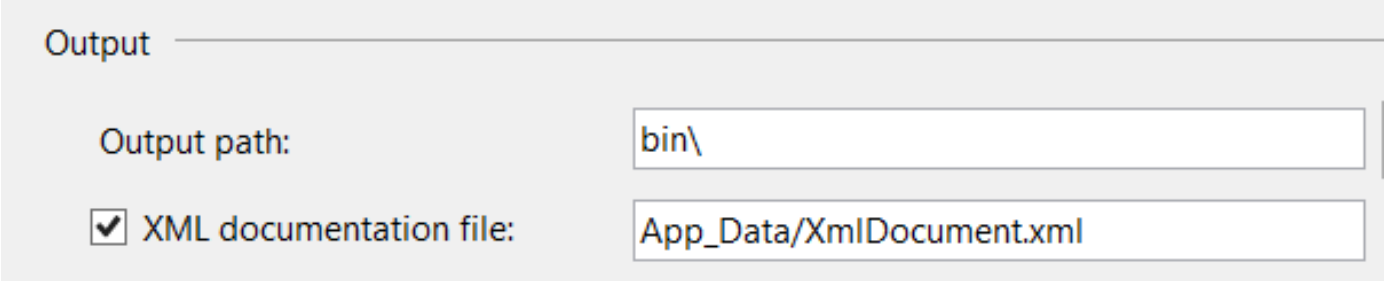
بصورت پیش فرض صفحات راهنما از placeholder-text برای مستندات استفاده می‌کنند. می‌توانید برای ساختن مستندات از توضیحات XML استفاده کنید. برای فعال سازی این قابلیت فایل Areas/HelpPage/App_Start/HelpPageConfig.cs را باز کنید و خط زیر را از حالت کامنت درآورید:

```
config.SetDocumentationProvider(new XmlDocumentationProvider(
    HttpContext.Current.Server.MapPath("~/App_Data/XmlDocument.xml")));
```

حال روی نام پروژه کلیک راست کنید و **Properties** را انتخاب کنید. در پنجره باز شده قسمت **Build** را کلیک کنید.



زیر قسمت **Output** گزینه **XML documentation file** را تیک بزنید و در فیلد روبروی آن مقدار "App_Data/XmlDocument.xml" را وارد کنید.



حال کنترلر ValuesController را از مسیر /Controllers/ValuesController.cs باز کنید و یک سری توضیحات XML به متدهای آن اضافه کنید. بعنوان مثال:

```
/// <summary>
/// Gets some very important data from the server.
/// </summary>
public IEnumerable<string> Get()
{
    return new string[] { "value1", "value2" };
}

/// <summary>
/// Looks up some data by ID.
/// </summary>
/// <param name="id">The ID of the data.</param>
public string Get(int id)
{
    return "value";
}
```

اپلیکیشن را مجددا اجرا کنید و به صفحات راهنما بروید. حالا مستندات API شما باید تولید شده و نمایش داده شوند.

صفحات راهنما مستندات شما را در زمان اجرا از توضیحات XML استخراج می‌کنند. دقت کنید که هنگام توزیع اپلیکیشن، فایل XML را هم منتشر کنید.

توضیحات تکمیلی

صفحات راهنما توسط کلاس **ApiExplorer** تولید می‌شوند، که جزئی از فریم ورک ASP.NET Web API است. به ازای هر API این کلاس یک **ApiDescription** دارد که توضیحات لازم را در بر می‌گیرد. در اینجا منظور از "API" ترکیبی از متدهای HTTP و مسیرهای نسبی است. بعنوان مثال لیست زیر تعدادی API را نمایش می‌دهد:

GET /api/products
GET /api/products/{id}
POST /api/products

اگر اکشن‌های کنترلر از متدهای متعددی پشتیبانی کنند، ApiExplorer هر متد را بعنوان یک API مجزا در نظر خواهد گرفت. برای مخفی کردن یک API از ApiExplorer کافی است خاصیت **ApiExplorerSettings** را به اکشن مورد نظر اضافه کنید و مقدار خاصیت *IgnoreApi* آن را به true تنظیم نمایید.

```
[ApiExplorerSettings(IgnoreApi=true)]
public HttpResponseMessage Get(int id) {  }
```

همچنین می‌توانید این خاصیت را به کنترلرها اضافه کنید تا تمام کنترلر از ApiExplorer مخفی شود.

کلاس ApiExplorer متن مستندات را توسط اینترفیس **IDocumentationProvider** دریافت می‌کند. کد مربوطه در مسیر /Areas/HelpPage/XmlDocumentation.cs قرار دارد. همانطور که گفته شد مقادیر مورد نظر از توضیحات XML استخراج می‌شوند. نکته جالب آنکه می‌توانید با پیاده سازی این اینترفیس مستندات خود را از منبع دیگری استخراج کنید. برای اینکار باید متد الحاقی **SetDocumentationProvider** را هم فراخوانی کنید، که در **HelpPageConfigurationExtensions** تعریف شده است.

کلاس **ApiExplorer** بصورت خودکار اینترفیس IDocumentationProvider را فراخوانی می‌کند تا مستندات APIها را دریافت کند. سپس مقادیر دریافت شده را در خاصیت **Documentation** ذخیره می‌کند. این خاصیت روی آبجکت‌های **ApiDescription** و **ApiParameterDescription** تعریف شده است.

مطالعه بیشتر

Adding a simple Test Client to ASP.NET Web API Help Page
Making ASP.NET Web API Help Page work on self-hosted services
Design-time generation of help page (or client) for ASP.NET Web API
Advanced Help Page customizations

نظرات خوانندگان

نویسنده: سعید شیرزادیان
تاریخ: ۱۳۹۲/۱۱/۱۸ ۱۹:۵۵

سلام؛ می‌خواستم بدونم قابلیت فوق نیز بر روی پروژه‌های asp.net وب فرمز نیز فعال می‌گردد؟ با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۸ ۲۰:۳۵

Enabling ASP.NET Web API Help Pages for ASP.NET Web Forms Applications