بسم اللّه الرحمن الرحیم

# آموزشگاه تحلیل داده

## تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

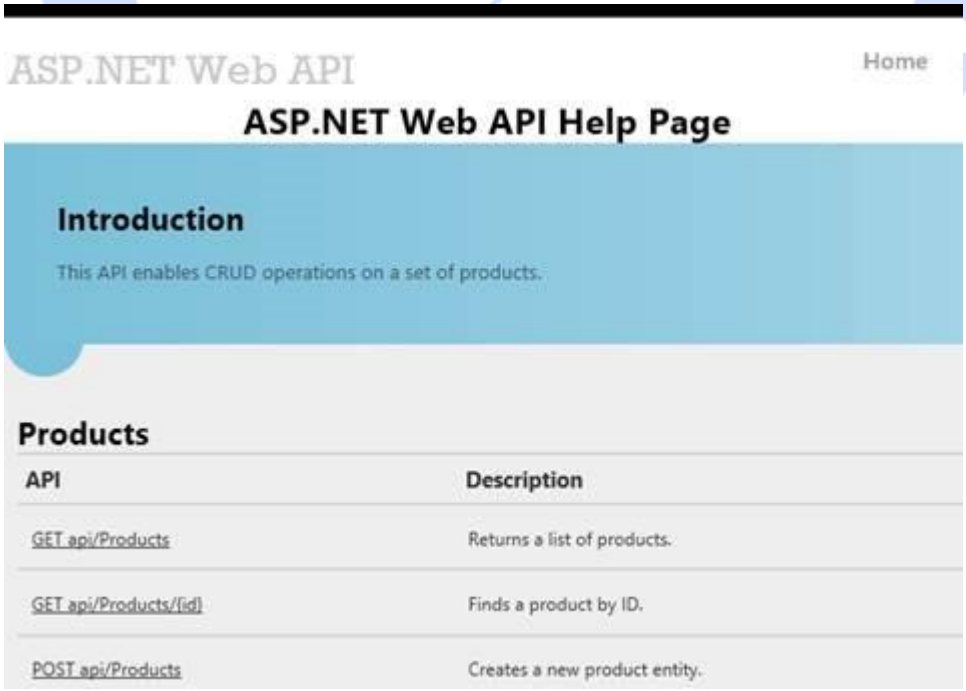
# ساختن صفحات Help برای Asp.Net Web API

مدرس : مهندس افشین رفوآ

دوره آموزشی Web API

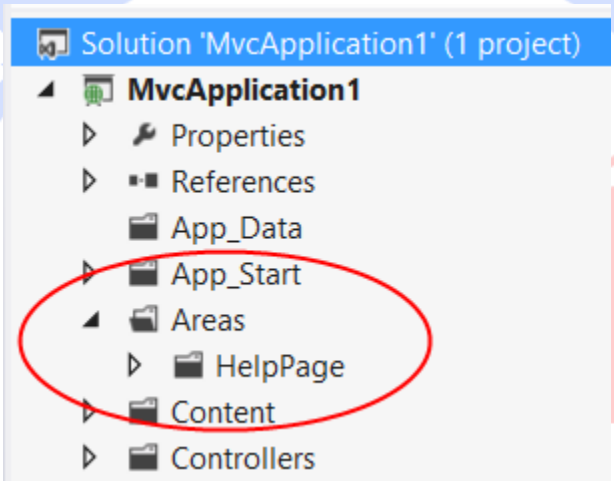
گاهی اوقات برای آنکه برنامه نویسان دیگر چگونگی فراخوانی شما را متوجه شوند، لازم است شما در زمان ساختن یک Web API، یک صفحه Help هم بسازید. شما می توانید تمام مستندات را به طور دستی بسازید ولی بهتر است تا جایی که امکان دارد از ساخت اتوماتیک(autogenerate) استفاده کنید.

برای فهم راحتتر این موضوع، Asp.net Web API یک کتابخانه برای ساخت اتوماتیک صفحات help در زمان اجرای برنامه در نظر گرفته است.



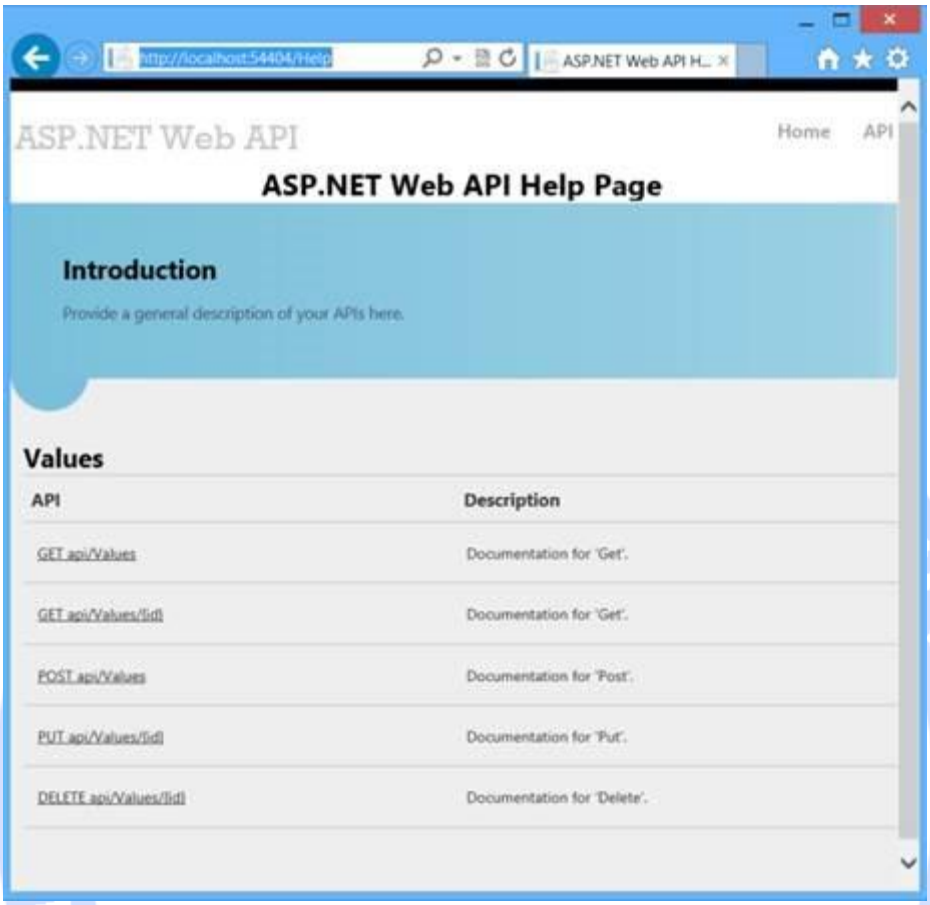
## ساختن صفحات API Help

ASP.NET and Web Tools 2012.2 Update را نصب کنید. این Update ، صفحات help را با پروژه های Web API ادغام می کند. سپس یک پروژه Asp.net MVC ایجاد کنید و پروژه Web API را انتخاب کنید. یک API Controller بسازید و نام آن را ValuesController قرار دهید. تمامی فایل های کد برای صفحه help در فولدر Area در پروژه قرار دارد.




وقتی شما برنامه را اجرا می کنید، به صفحه اصلی برنامه یک لینک API اضافه می شود.


این لینک شما را به صفحه خلاصه API منتقل می کند.




View مربوط به این صفحه در MVC در Area به صورت Area/HelpPage/Views/Help/Index.cshhtml تعریف می شود. شما می توانید این صفحه را برای تغییر دادن طرح و ترکیب، معرفی، موضوع، سبک ها و غیره ویرایش کنید.

بخش اصلی صفحه جدول API ها بر اساس Controller ها گروه بندی شده است. محتوای جدول به صورت پویا توسط رابط IApiExplorer ساخته می شود(در ادامه بیشتر به بررسی این رابط می پردازیم).اگر شما یک API Controller جدید را اضافه کنید، جدول در زمان اجرا به صورت خودکار به روز خواهد شد.

ستون API، متد های HTTP و URI نسبی را لیست می کند. ستون Description هم شامل مستنداتی برای هر API است. در ابتدا، Documentation صرفا یک مکان نگه دارنده متن (placeholder text) می باشد.در ادامه به شما نشان خواهیم داد چگونه XML را اضافه کنیم.

هر API به یک صفحه برای جزئیات بیشتر که شامل مثالی از درخواست و پاسخ می باشد، یک لینک دارد.


## اضافه کردن صفحات Help به یک پروژه موجود

شما می توانید با استفاده از گزینه NuGet Package Manager به یک پروژه Web API موجود صفحات help اضافه کنید. این گزینه زمانی که شما از یک template پروژه دیگر غیر از Web API template استفاده می کنید مناسب است.

در منو Tools، گزینه Library Package Manager و سپس Package Manager Console را انتخاب کنید. در پنجره Package Manager Console یکی از مقادیر زیر را تایپ کنید.

برای برنامه c# : Install-Package Microsoft.AspNet.WebApi.HelpPage

برای برنامه Visual Basic: Install-Package Microsoft.AspNet.WebApi.HelpPage.VB

این کد های بالا ، اسمبلی های ضروری را نصب می کنند و view های MVC برای صفحات help(که در فولدر Areas/HelpPage مسیر داده شده است) را اضافه می کند. برای ساختن یک لینک در razor view، کد زیر را وارد کنید:

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

همچنین مطمئن باشید که areaها رجیستر شده باشد. در global.asax اگر کد زیر موجود نبود، آن را به متد Application_Start اضافه کنید:

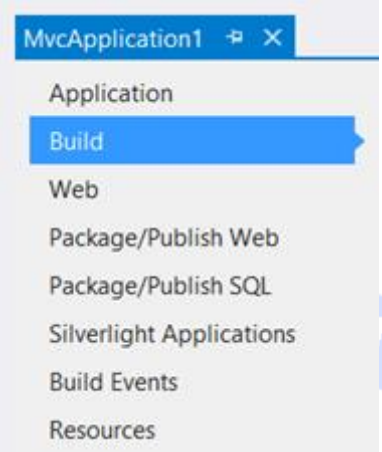
```
protected void Application_Start()
{
   // Add this code, if not present.
   AreaRegistration.RegisterAllAreas();

   // ...
}
```

## اضافه کردن مستندات(Documentation) API

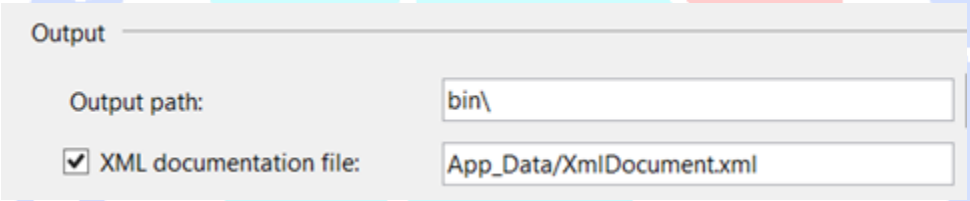
به صورت پیش فرض، صفحات help برای مستندات یک مکان نگه دارنده رشته ای( placeholder string) وجود دارد. شما می توانید از XML برای ساخت Documentation استفاده کنید. برای فعالسازی این ویژگی، به Areas/HelpPage/App_Start/HelpPageConfig.cs بروید و خط زیر را از حالت کامنت بودن خارج کنید:

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

حالا برای فعالسازی XML Documentation، به solution explorer بروید و روی پروژه کلیک راست کنید و در قسمت Properties گزینه Build را انتخاب کنید:




در زیر Output، گزینه XML documentation file را انتخاب کنید و مقدار آن را App_Data/XmlDocument.xml قرار دهید.




سپس، یک ValueController برای ControllerAPI که به صورت /Controller/ValueControler.cs می باشد، باز کنید. بعضی از کامنت های Documentation متد های Controller را اضافه کنید. برای مثال:

```
///
/// Gets some very important data from the server.
///
public IEnumerable<string>Get()
{
   return new string[] { "value1", "value2" };
}
///
/// Looks up some data by ID.
///
/// The ID of the data.
public string Get(int id)
{
   return "value";
}
```

حالا build کنید و برنامه را دوباره اجرا کنید. رشته های Documentation در جدول API ظاهر می شود.

| API | Description |
| --- | --- |
| GET api/Values | Gets some very important data from the server. |
| GET api/Values/{id} | Looks up some data by ID. |

صفحه help رشته ها را از فایل XML در زمان اجرا می خواند(هرگاه برنامه خود را پیاده سازی کردید مطمئن شوید که فایل XML هم پیاده سازی شده باشد.)

صفحات help بالای کلاس ApiExplorer ساخته شده است که قسمتی از Web API framework می باشد. کلاس ApiExplorer مواد اولیه برای ساختن یک صفحه help محیا می سازد. هر API، ApiExplorer شامل یک ApiDescription می باشد که API را توضیح می دهد. برای این منظور، یک API ترکیبی از متد HTTP و URI نسبی تعریف شده است. برای مثال چند نمونه از API های متمایز را در زیر می بینید:

- GET /api/Products
- GET /api/Products/{id}
- POST /api/Products

اگر یک action کنترولر از چندین متد HTTP پشتیبانی کند، ApiExplorer با هر متد به صورت یک API متمایز رفتار می کند. برای مخفی سازی یک API از ApiExplorer ، ApiExplorerSettings Attribute را به action اضافه کنید و مقدار IgnoreApi را true قرار دهید.

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

شما همچنین برای اعمال بر روی کل Controller می توانید این Attribute را بالای Controller اضافه کنید.

کلاس ApiExplorer مستندات رشته ای خود را از IDocumentionProvider دریافت می کند. همانطور که قبلا مشاهده کردید، کتابخانه صفحات help، یک IDocumentionProvider آماده می کند که Documentation را از

Documentation XML دریافت می کند. مکان کد ها را می توان از آدرس /Areas/HelpPage/XmlDocumentationProvider.cs دنبال کرد. شما همچنین می توانید Documentation را از منابع دیگر با نوشتن IDocumentionProvider دلخواه، دریافت کنید. برای دسترسی به آن، متد SetDocumentationProvider را که در HelpPageConfigurationExtensions تعریف شده است، فراخوانی کنید.ApiExplorer به طور خودکار IDocumentionProvider را برای دریافت Documentation رشته ای برای هر Api فراخوانی می کند