

ASP.NET Web API 2 به همراه یک سری قابلیت جدید جالب منتشر شده است. در این پست 5 قابلیت برتر از این قابلیت‌های جدید را بررسی می‌کنیم.

1. Attribute Routing

در کنار سیستم routing فعلی، ASP.NET Web API 2 حالا از Attribute Routing هم پشتیبانی می‌کند. در مورد سیستم routing فعلی، می‌توانیم قالب‌های متعددی برای routing بنویسیم. هنگامی که یک درخواست به سرور میرسد، کنترلر مناسب انتخاب شده و اکشن متد مناسب فراخوانی می‌شود. در لیست زیر قالب پیش فرض routing در Web API را مشاهده می‌کنید.

```
Config.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{Controller}/{id}",
    defaults: new { id = RouteParameter.Optional }
);
```

این رویکرد routing مزایای خود را دارد. از جمله اینکه تمام مسیرها در یک مکان واحد تعریف می‌شوند، اما تنها برای الگوهای مشخص. مثلاً پشتیبانی از nested routing روی یک کنترلر مشکل می‌شود. در ASP.NET Web API 2 به سادگی می‌توانیم الگوی URI ذکر شده را پشتیبانی کنیم. لیست زیر نمونه ای از یک الگوی URI با AttributeRouting را نشان می‌دهد.

URI Pattern --> books/1/authors

```
[Route("books/{bookId}/authors")]
public IEnumerable<Author> GetAuthorByBook(int bookId) { ..... }
```

2. CORS - Cross Origin Resource Sharing

بصورت نرمال، مرورگرها اجازه درخواست‌های cross-domain را نمی‌دهند، که بخاطر same-origin policy است. خوب، CORS (Cross Origin Resource Sharing) چیست؟ CORS یک مکانیزم است که به صفحات وب این را اجازه می‌دهد تا یک درخواست آژاکسی (Ajax Request) به دامنه ای دیگر ارسال کنند. دامنه ای به غیر از دامنه ای که صفحه وب را رندر کرده است. CORS با استانداردهای W3C سازگار است و حالا ASP.NET Web API در نسخه 2 خود از آن پشتیبانی می‌کند.

3. OWIN (Open Web Interface for .NET) self-hosting

ASP.NET Web API 2 به همراه یک پکیج عرضه می‌شود، که *Microsoft.AspNet.WebApi.OwinSelfHost* نام دارد.

طبق گفته وب سایت <http://owin.org>:

OWIN یک اینترفیس استاندارد بین سرورهای دات نت و اپلیکیشن‌های وب تعریف می‌کند. هدف این اینترفیس جداسازی (decoupling) سرور و اپلیکیشن است. تشویق به توسعه ماژول‌های ساده برای توسعه اپلیکیشن‌های وب دات نت. و بعنوان یک استاندارد باز (open standard) اکوسیستم نرم افزارهای متن باز را تحریک کند تا ابزار توسعه اپلیکیشن‌های وب دات نت توسعه یابند.

بنابراین طبق گفته‌های بالا، OWIN گزینه ای ایده آل برای میزبانی اپلیکیشن‌های وب روی پروسس‌هایی به غیر از پروسس IIS است. پیاده سازی‌های دیگری از OWIN نیز وجود دارند، مانند Giacomo, Kayak, Firefly و غیره. اما *Katana* گزینه توصیه شده برای سرورهای مایکروسافت و فریم ورک‌های Web API است.

4. IHttpActionResult

در کنار دو روش موجود فعلی برای ساختن response اکشن متدها در کنترلر ها، ASP.NET Web API 2 حالا از مدل جدیدی هم

پشتیبانی می‌کند. *IHttpResponseMessage* یک اینترفیس است که بعنوان یک فاکتوری (factory) برای *HttpResponseMessage* کار می‌کند. این روش بسیار قدرتمند است بدلیل اینکه web api را گسترش می‌دهد. با استفاده از این رویکرد، می‌توانیم response هایی با هر نوع دلخواه بسازیم. برای اطلاعات بیشتر به [how to serve HTML with IHttpActionResult](#) مراجعه کنید.

5. Web API OData

پروتکل OData (Open Data Protocol) در واقع یک پروتکل وب برای کوئری گرفتن و بروز رسانی داده‌ها است. ASP.NET Web API 2 پشتیبانی از *\$expand*, *\$select* و *\$value* را اضافه کرده است. با استفاده از این امکانات، می‌توانیم نحوه معرفی پاسخ سرور را کنترل کنیم، یعنی representation دریافتی از سرور را می‌توانید سفارشی کنید. **\$expand**: بصورت نرمال، هنگام کوئری گرفتن از یک کالکشن OData، پاسخ سرور موجودیت‌های مرتبط (related entities) را شامل نمی‌شود. با استفاده از *\$expand* می‌توانیم موجودیت‌های مرتبط را بصورت inline در پاسخ سرور دریافت کنیم. **\$select**: از این متد برای انتخاب چند خاصیت بخصوص از پاسخ سرور استفاده می‌شود، بجای آنکه تمام خاصیت‌ها بارگذاری شوند. **\$value**: با این متد مقدار خام (raw) فیلدها را بدست می‌آورید، بجای دریافت آنها در فرمت OData.

چند مقاله خوب دیگر

[Top 10 ASP.NET MVC Interview Questions](#)

[jQuery code snippets every web developer must have 7](#)

[WCF Vs ASMX Web Services](#)

[Top 10 HTML5 Interview Questions](#)

[JavaScript : Validating Letters and Numbers](#)