بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

ردیابی (Tracing) در ASP.NET Web API 2

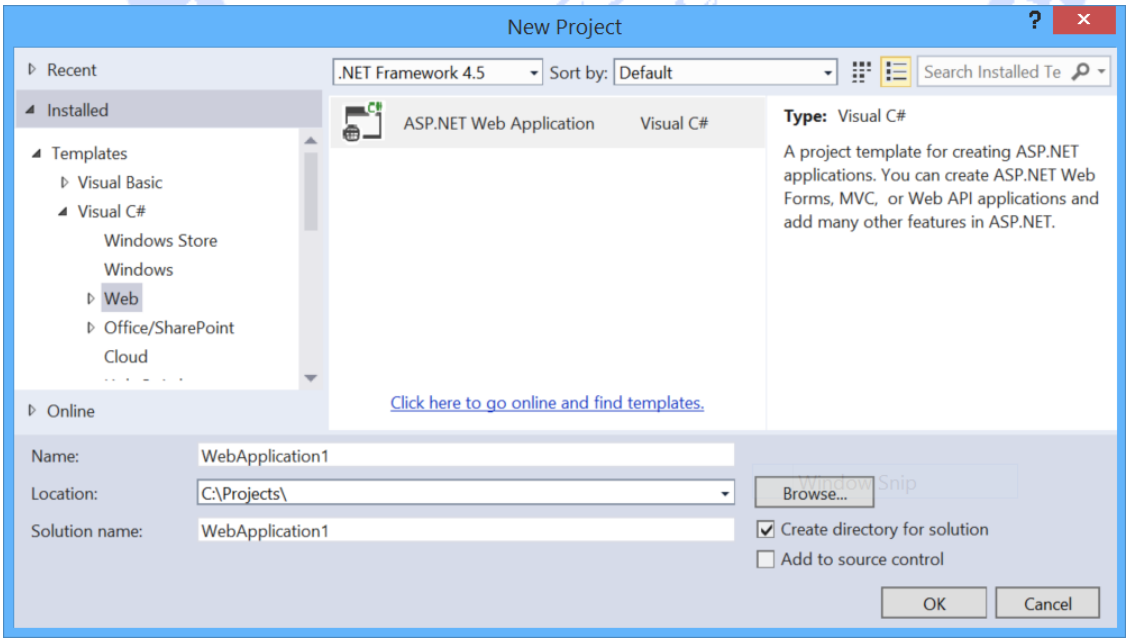
مدرس : مهندس افشین رفوآ

دوره آموزشی Web API

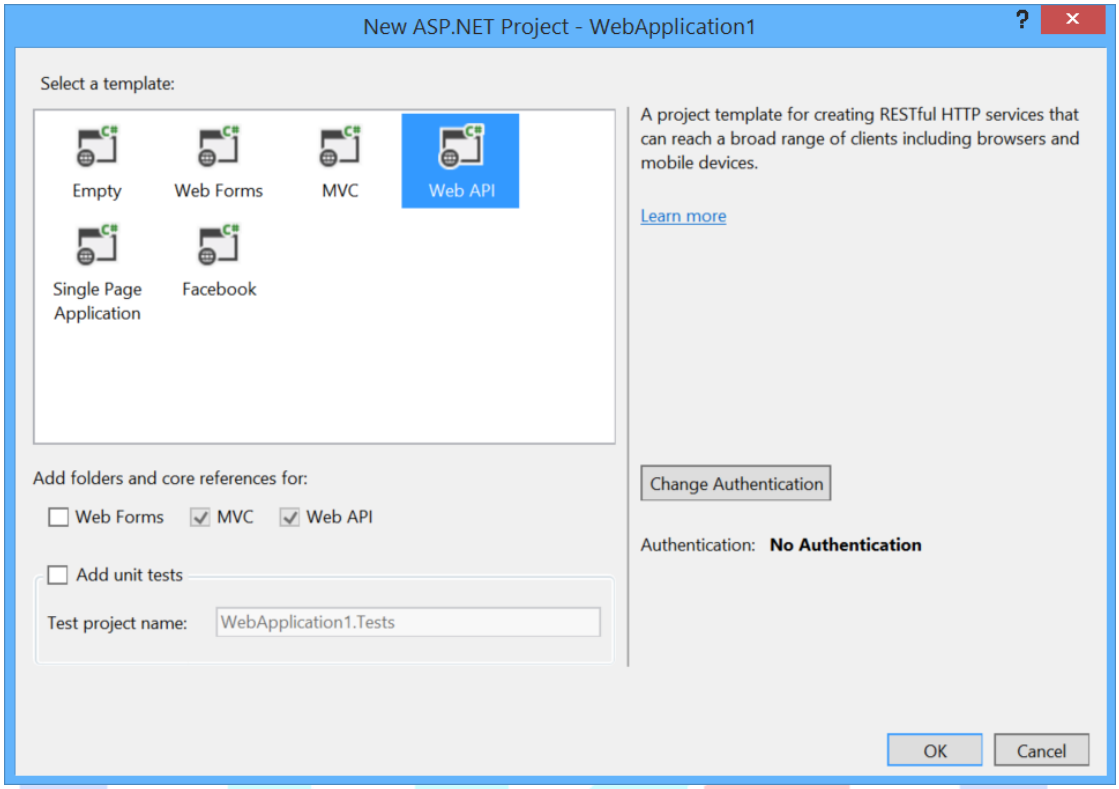
زمانی که شما می خواهید یک برنامه تحت وب را اشکال زدایی(debug) کنید هیچ جایگزینی برای یک مجموعه خوب از trace کردن log ها وجود ندارد. در این مقاله قصد داریم چگونگی trace کردن در ASP.NET Web API را به شما آموزش دهیم.

فعال کردن trace کردن system.diagnostic‌ها در Web API

ابتدا ما یک پروژه ASP.Net Web API می سازیم . برای این کار ابتدا visual studio را باز کنید و در منو File گزینه New و سپس بر روی Project کلیک کنید. حالا ASP.NET Web Application را انتخاب کنید.



در پنجره زیر Web API را انتخاب کنید.



در منو Tools گزینه Library Package Manager و سپس Package Manage Console را انتخاب کنید. در Package Manager Console دستورات زیر را تایپ کنید.

```
Install-Package Microsoft.AspNet.WebApi.Tracing
Update-Package Microsoft.AspNet.WebApi.WebHost
```

اولین خط کد بالا آخرین ورژن Web API Tracing را نصب می کند. همچنین هسته بسته های Web API را به روز رسانی می کند. دومین خط نیز بسته WebApi.WebHost را به آخریم ورژن به روز رسانی می کند.

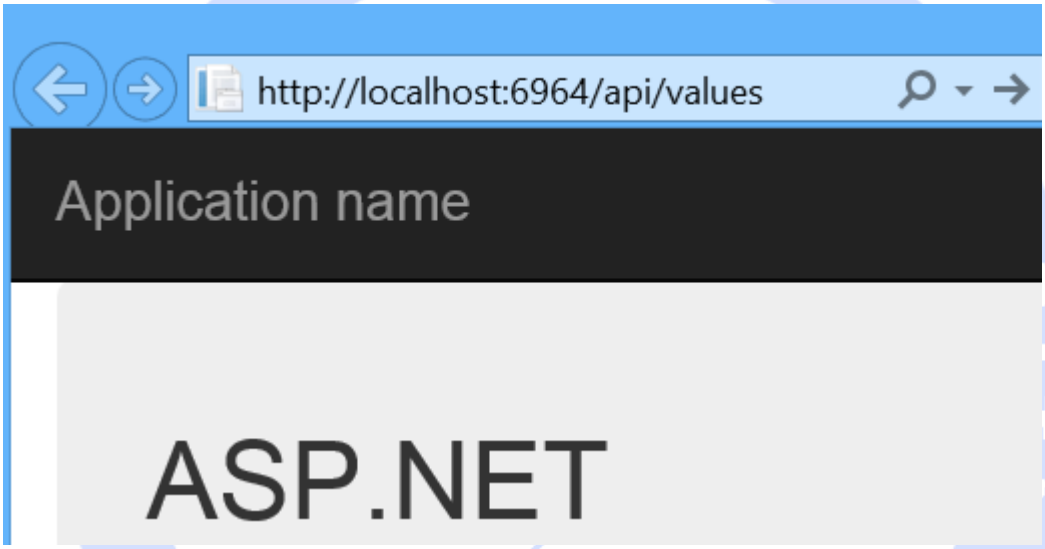
نکته: اگر می خواهید ورژن Web API خود را روی یک ورژن خاص مشخص کنید، Version- را در زمان نصب بسته tracing به انتهای کد اضافه کنید و ورژن مورد نظر خود را بعد از آن بنویسید.

در پوشه App_Start فایل WebApiConfig.cs را باز کنید و کد زیر را به متد Register آن اضافه کنید.

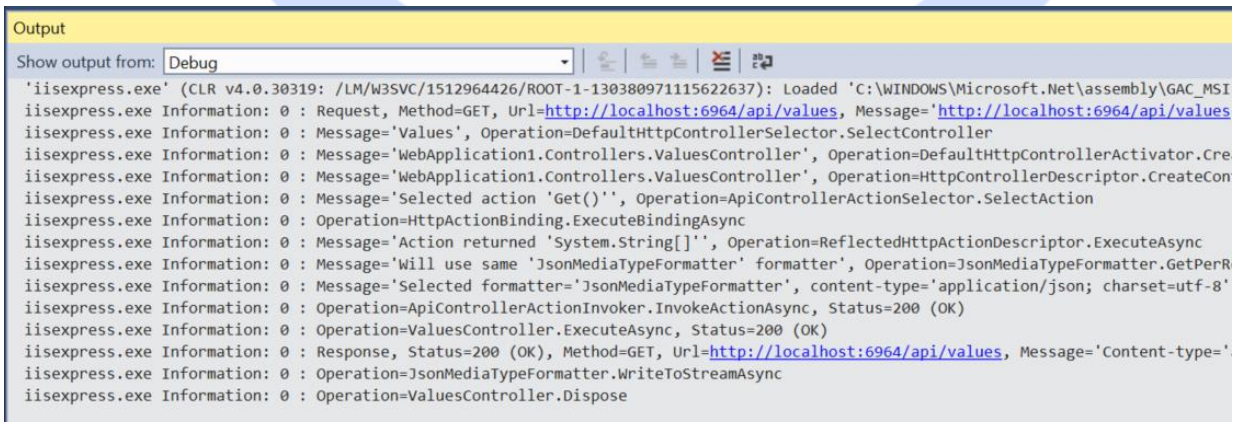
```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        // New code
        config.EnableSystemDiagnosticsTracing();
        // Other configuration code not shown.
```

```
    }
}
```

در کد بالا، کلاس SystemDiagnosticsTraceWriter را به خط لوله(Pipeline) اضافه کنید. کلاس SystemDiagnosticsTraceWriter مربوط به traceها را در System.Diagnostics.Trace می نویسیم. برای دیدن trace ها برنامه را debuge کنید و برنامه را اجرا و اگر /api/values در قسمت آدرس قرار نداشت، آن را اضافه کنید.



حالات trace در پنجره output در visual studio نشان داده می شود.(در منو View گزینه Output را انتخاب کنید.)



SystemDiagnosticsTraceWriter، traceها را در System.Diagnostics.Trace می نویسد اما شما می توانید مکان trace دیگری مشخص کنید، برای مثال می توانید traceها را در یک فایل log بنویسید.

پیکره بندی SystemDiagnoticsTraceWriter

کد زیر به شما نشان می دهد چگونه trace writer را طراحی کنید.

```
SystemDiagnosticsTraceWriter traceWriter = config.EnableSystemDiagnosticsTracing();
traceWriter.IsVerbose = true;
traceWriter.MinimumLevel = TraceLevel.Debug;
```

شما می توانید دو مورد را کنترل کنید:

- IsVerbose: اگر false باشد، هر trace شامل حداقل اطلاعات می باشد. اگر true باشد، trace شامل اطلاعات اضافی می باشد.
- minimumLevel: کمترین مقدار trace را قرار می دهد. مقادیر trace به ترتیب ، اشکال زدایی(debug)، اطلاعات(Info)، هشدار (Warn)، خطا (Error) و وخیم (Fatal) هستند.

اضافه کردن trace ها در برنامه Web API

اضافه کردن trace writer یک دسترسی سریع به trace هایی که با Web API pipeline ساخته شده است می دهد. شما همچنین می توانید از trace writer برای کد های خودتان استفاده کنید:

```
using System.Web.Http.Tracing;
public class ProductsController : ApiController
{
    public HttpResponseMessage GetAllProducts()
    {
        Configuration.Services.GetTraceWriter().Info(
            Request, "ProductsController", "Get the list of products.");
        // ...
    }
}
```

برای دسترسی به trace writer، متد HttpConfiguration.Services.GetTraceWriter را فراخوانی کنید. در یک Controller، این متد از طریق ApiController.Configuration قابل دسترس می باشد.

برای نوشتن یک trace، شما می توانید متد ITraceWriter.Trace را مستقیما فراخوانی کنید اما کلاس ITraceWriterExtensions چند extension method که آسان پسند تر(more friendly) باشند تعریف می کند.

زیر ساخت trace کردن Web API

در این قسمت چگونگی نوشتن یک trace writer سفارشی در Web API را توضیح می دهیم.

بسته ی Microsoft.AspNet.WebApi.Tracing بیشترین استفاده را در زیرساخت های trace کردن Web API دارد. به غیر از استفاده از Microsoft.AspNet.WebApi.Tracing شما می توانید از کتابخانه های دیگر trace کردن یا log گرفتن استفاده کنید مثل Nlog یا log4net.

برای گردآوری trace ها می توانید رابط ITraceWriter را پیاده سازی کنید. یک مثال ساده را در زیر می بینید:

```
public class SimpleTracer : ITraceWriter
{
    public void Trace(HttpRequestMessage request, string category, TraceLevel level,
        Action<TraceRecord> traceAction)
    {
        TraceRecord rec = new TraceRecord(request, category, level);
        traceAction(rec);
        WriteTrace(rec);
    }
    protected void WriteTrace(TraceRecord rec)
    {
        var message = string.Format("{0};{1};{2}",
            rec.Operator, rec.Operation, rec.Message);
        System.Diagnostics.Trace.WriteLine(message, rec.Category);
    }
}
```

متد ITraceeWriter.Trace یک trace می سازد.فراخوانی کننده یک category و رتبه(level) trace مشخص می کند.Category می تواند هر رشته ای که کاربر تعریف می کند باشد. پیاده سازی trace شما باید اعمال زیر را انجام دهد:

۱ – یک TraceRecord جدید بسازد. همانطور که نشان داده شد درخواست(request)،رده(category) و رتبه trace را مقداردهی اولیه کند. این مقادیر می تواند توسط فراخوانی کننده فراهم گردد.

۲– نماینده(delegate) traceAction را فراخوانی کند. در این delegate، انتظار می رود فراخوانی کننده TraceRecord را پر کند.

۳- با استفاده از هر تکنیک log گرفتن که بخواهید، TraceRecord را بنویسید. مثالی که در اینجا نشان داده شده به سادگی System.Diagnostics.Trace را فراخوانی می کند.

تنظیم کردن trace writer

برای فعالسازی trace کردن، شما باید Web API را برای استفاده از پیاده سازی ITraceWriter خودتان تنظیم کنید. برای انجام این کار از طریق HttpConfiguration می توانید به صورت کد های زیر عمل کنید:

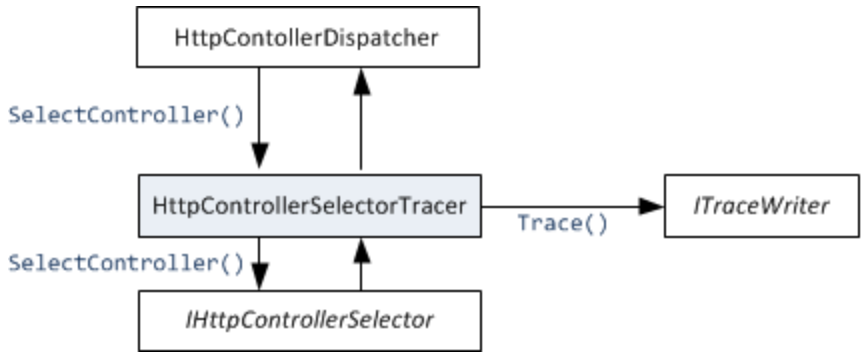
```
public static void Register(HttpConfiguration config)
{
    config.Services.Replace(typeof(ITraceWriter), new SimpleTracer());
}
```

تنها یک trace writer می تواند فعال باشد. به صورت پیش فرض، Web API یک trace کننده no-op که کاری انجام نمی دهد اضافه می کند. با توجه به اینکه Trace کننده no-op وجود دارد بنابراین فرآیند trace کردن لازم نیست null بودن trace writer را قبل از نوشتن چک کنند.

چگونگی کارکرد trace کردن Web API

Trace کردن Web API از الگوی facade استفاده می کند: هرگاه trace کردن فعال شود، Web API قسمت های مختلفی از درخواست pipeline با کلاس هایی که trace انجام می دهد فرا می خواند.

برای مثال، هرگاه یک controller انتخاب می شود، pipeline از رابط IHttpControllerSelector استفاده می کند. trace کردن، pipeline یک کلاس که IHttpControllerSelector را پیاده سازی می کند وارد می کند اما از طریق پیاده سازی واقعی فراخوانده می شود:



مزایای این طراحی عبارت اند از:

اگر شما trace writer را اضافه نکنید، component های trace کردن هیچ تاثیر خاصی ندارد.

اگر شما سرویس پیش فرضی مثل IHttpControllerSelector را با پیاده سازی سفارشی خودتان عوض کنید trace کردن تاثیری ندارد زیرا trace کردن با wrapper انجام می گیرد.

شما می توانید کل فریم ورک Web API trace را به وسیله سرویس پیش فرض ITraceManager با فریم ورک سفارشی خودتان عوض کنید.

```
config.Services.Replace(typeof(ITraceManager), new MyTraceManager());
```

برای مقداردهی اولیه سرویس trace خودتان، ITraceManager.Initialize را پیاده سازی کنید. به یاد داشته باشید که این تعویض فریم ورک entire trace، شامل همه ی کد های trace کردن که در Web API ساخته شده می شود.