

Capturing data from a .Net web-service that fails with an HTTP 500 error code

Asked 16 years, 2 months ago Modified 6 years, 2 months ago Viewed 4k times



4



I have a .net web-service hosted in IIS 6.0 that periodically fails with an http 500 because a client connects to it with data that does not match the wsdl.

Things like having an element specified in a method as being of type int and the inbound xml element contains a decimal number.



WSDL element definition:

```
<s:element minOccurs="1"
  maxOccurs="1"
  form="unqualified" name="ItemCount" type="s:int" >
```

provided element:

```
<ItemCount>1.0</ItemCount>
```

This leaves a 500 error in the iis logs but no information of the soap fault returned or the input data that caused the error.

Currently I have diagnosed several problems with data provided by capturing everything using wireshark but I'd like to know of other options that are perhaps less intrusive.

Is there any way of capturing the data being sent that is causing the 500 errors (hopefully ONLY capturing the data when the 500 occurs)? Possibly by:

- Configuring IIS
- Configuring the web-service
- Changing the code of the web-service

EDIT after testing answer provided by tbreffni

The answer that best fitted what I was after tbreffni's - there were several other good responses but the answer allows capture of the payload causing deserialization errors without running something like fiddler or wireshark.

Info on actually getting a SOAP extension to run is a little light so I've included below the steps I found necessary:

- Build the SOAP extension as a .dll as per the [MSDN](#) article
- Add the .dll to the bin directory for the service to trace
- In the web.config of the service to trace add the following to the webServices section, replacing the SOAPTraceExtension.TraceExtension and SOAPTraceExtension to match your extension.

```
<webServices>
  <soapExtensionTypes>
    <add type="SOAPTraceExtension.TraceExtension, SOAPTraceExtension"
      priority="1" group="0"/>
  </soapExtensionTypes>
</webServices>
```

.net

web-services

iis

monitoring

trace

Share

edited Oct 22, 2018 at 19:34

Improve this question

Follow

asked Oct 16, 2008 at 21:18



David Hall

33.1k ● 10 ● 92 ● 129

Are you capturing cs-uri-query in your IIS logs? securityfocus.com/infocus/1712

– [Solracnapod](#) Oct 20, 2008 at 23:54

6 Answers

Sorted by: Highest score (default)





I'd try [fiddler](#) or [here](#). Not specifically for web services and typically for client side, it can be used as a reverse [proxy](#).

3



It's very script-able and "request/response" aware so I think it likely you could get it to only capture 500 errors.



Share

edited Feb 27, 2013 at 10:56

answered Oct 16, 2008 at 21:54



Improve this answer



Liam

29.4k ● 28 ● 137 ● 199



Tony Lee

5,632 ● 1 ● 30 ● 47

Follow

Thanks - I have been using wireshark (in a lot of ways like fiddler on steroids) but was hoping for something a little less intrusive - the badly formed xml is being sent to a production machine. – [David Hall](#) Oct 16, 2008 at 22:03



You could implement a global exception handler in your web service that logs the details of any exceptions that occur. This is useful for your current problem, plus it's very useful in a production environment as it gives you an insight into how many exceptions are being thrown and by what code.

2



To implement an exception handler for a .Net web service, you need to create a SOAP extension. See the following [MSDN Article](#) for an example. I've used this approach in several production web services, and it's been invaluable in determining what issues are occurring and where.



Share Improve this answer Follow

answered Oct 20, 2008 at 23:45



tbreffni

5,132 ● 5 ● 32 ● 30



1



I never used this myself, but I just found this in MSDN: [Enabling Tracing in ASP.NET Web Services](#)

Share Improve this answer Follow

answered Oct 16, 2008 at 22:34



Quantenmechaniker

514 ● 3 ● 6

Useful! - it gives good info including the soap fault Interestingly, it logs warnings for unexpected tags, and says what the bad element is but gives no detail on data that causes exceptions in valid tags Upvoting and not marking as answer - it doesn't seem to trace input. I'll keep looking! – [David Hall](#) Oct 16, 2008 at 23:40



1



Have you looked at IIS Request-Based Tracing?: <http://technet.microsoft.com/en-us/library/cc786920.aspx>

Share Improve this answer Follow

answered Oct 21, 2008 at 0:00



Tony Lee

5,632 ● 1 ● 30 ● 47



0



After some thinking of my own, the current best solution I can see is to throw together a light weight facade web-service that accepts an xml blob as input.

I can then get my facade service to call the real service with the input data provided by the client and then:

- handle any errors, logging the source data and the returned soap fault
- Pass back to the client the responses from good data

This would only be a temporary measure (I'm strongly opposed to web-services that are not explicit about the XML they accept) but it would perhaps give me more leverage to get over the hump of diagnosing errors like this in production where the client connecting to the web-service is not honouring the wsdl or able to read soap faults returned.

Thankfully in this case there is only one party posting to the web-service - the packet sniffer methods (fiddler or wireshark) are feasible but the lack of logging for 500 errors did get me thinking "what nicer options are there?".

Share Improve this answer Follow

answered Oct 16, 2008 at 22:14



David Hall

33.1k ● 10 ● 92 ● 129

Have you looked at IIS Request-Based Tracing?: technet.microsoft.com/en-us/library/cc786920.aspx – Tony Lee Oct 16, 2008 at 22:16

Please, post that as an answer, I'd like to vote it up, I've never heard of it before. Don't know if it will address my particular problem but even if it doesn't, it is good to know about.

– David Hall Oct 16, 2008 at 22:23



If you want a use a tool, use WireShark.

0

Otherwise, the easiest way is to capture "everything" (assuming you are using ASMX) is to enable trace on system.net



<http://blogs.msdn.com/dgorti/archive/2005/09/18/471003.aspx>



If you are using WCF, it has great tracing support and you can use svctraceviewer to view the trace logs. Doesn't sound like you are using WCF though.



Share Improve this answer Follow

answered Oct 20, 2008 at 23:57



jezell

2,532 ● 14 ● 12
