

# Exceptions in Web Services

Asked 16 years, 3 months ago   Modified 6 years, 7 months ago

Viewed 1k times



2



My group is developing a service-based (.NET WCF) application and we're trying to decide how to handle exceptions in our internal services. Should we throw exceptions? Return exceptions serialized as XML? Just return an error code?



Keep in mind that the user will never see these exceptions, it's only for other parts of the application.



.net

wcf

web-services

exception

Share

Improve this question

Follow

asked Sep 2, 2008 at 17:38



brien

4,430 ● 4 ● 31 ● 32

5 Answers

Sorted by:

Highest score (default)





3

WCF uses `SoapFaults` as its native way of transmitting exceptions from either the service to the client, or the client to the service.



You can declare a custom SOAP fault using the `FaultContract` attribute in your contract interface:



For example:



```
[ServiceContract(Namespace="foobar")]
interface IContract
{
    [OperationContract]
    [FaultContract(typeof(CustomFault))]
    void DoSomething();
}

[DataContract(Namespace="Foobar")]
class CustomFault
{
    [DataMember]
    public string error;

    public CustomFault(string err)
    {
        error = err;
    }
}

class myService : IContract
{
    public void DoSomething()
    {
        throw new FaultException<CustomFault>( new CustomFault(""));
    }
}
```

Share Improve this answer

edited May 18, 2018 at 9:20

Follow



ColdFire

6,859 ● 6 ● 36 ● 53

answered Sep 2, 2008 at 17:59



FlySwat

175k ● 75 ● 248 ● 314



2



Well, why not just throw the standard SOAPExceptions? The problem with error codes and serialized XML is that they both require additional logic to recognize that an error did in fact happen. Such an approach is only useful if you have specialized logging or logic that needs to happen on the other side of the web service. Such an example would be returning a flag that says "it's ok to continue" with an error exception report.

Regardless of how you throw it, it won't make the job any easier, as the calling side still needs to recognize there was an exception and deal with it.

Share Improve this answer

answered Sep 2, 2008 at 17:51

Follow



hova

2,841 ● 20 ● 19



1

I'm a bit confused, I'm not being flippant -- you say you want to return exceptions serialised as XML on the one hand and that the user will never see the exceptions on the other hand. Who will be seeing these exceptions?



Normally I'd say to use WCF fault contracts.



Share Improve this answer

answered Sep 2, 2008 at 17:47

Follow



[Phil Bennett](#)

4,839 ● 4 ● 31 ● 28



0

Phil, different parts of the application call each other using WCF. By "return exceptions serialized as XML," I meant that the return value of the function would be an exception object. Success would be indicated by null.



I don't think that's the right option.



WCF fault contracts sound good, but I don't know anything about them. Checking google right now.

Share Improve this answer

answered Sep 2, 2008 at 17:55

Follow



[brien](#)

4,430 ● 4 ● 31 ● 32



0

I would avoid sending exceptions directly back to the client unless you are okay with that much detail being sent back.



I would recommend using WCF faults to transmit your error message and code (something that can be used to make a decision on the receiver to retry, error out, etc) depending if it is the sender or receiver at fault.

This can be done using  
`FaultCode.CreateReceiverFaultCode` and  
`FaultCode.CreateSenderFaultCode`.

I'm in the process of going through this right now, but ran into a nasty snag it seems in the WCF fault generated SOAP 1.1 response. If you are interested, you can check out my question about it here:

[.NET WCF faults generating incorrect SOAP 1.1 faultcode values](#)

Share Improve this answer

Follow

edited May 23, 2017 at 10:32



Community Bot

1 • 1

answered Sep 15, 2008 at 17:47



wojo

11.8k • 4 • 24 • 21