

Is it possible to return objects from a WebService?

Asked 16 years, 4 months ago Modified 15 years, 11 months ago

Viewed 6k times



6



Instead of returning a common string, is there a way to return classic objects? If not: what are the best practices?

Do you transpose your object to xml and rebuild the object on the other side? What are the other possibilities?

web-services



Share

Improve this question

Follow

edited Aug 23, 2008 at 18:41



Chris

6,842 ● 6 ● 53 ● 67

asked Aug 15, 2008 at 1:51



Paul

2,833 ● 2 ● 19 ● 12

11 Answers

Sorted by:

Highest score (default)



7

As mentioned, you can do this in .net via serialization. By default all native types are serializable so this happens automatically for you.



However if you have complex types, you need to mark the object with the [Serializable] attribute. The same goes with complex types as properties.



So for example you need to have:



```
[Serializable]
public class MyClass
{
    public string MyString {get; set;}

    [Serializable]
    public MyOtherClass MyOtherClassProperty {get;
set;}
}
```

Share Improve this answer

answered Aug 15, 2008 at 2:23

Follow



[lomaxx](#)

116k ● 58 ● 147 ● 180



5

If the object can be serialised to XML and can be described in WSDL then yes it is possible to return objects from a webservice.



Share Improve this answer

answered Aug 15, 2008 at 1:57

Follow



[Kev](#)

120k ● 53 ● 305 ● 390



Yes: in .NET they call this serialization, where objects are serialized into XML and then reconstructed by the

3

consuming service back into its original object type or a surrogate with the same data structure.



Share Improve this answer

answered Aug 15, 2008 at 1:56



Follow



[Jon Limjap](#)

95.3k ● 15 ● 103 ● 153



2

Where possible, I transpose the objects into XML - this means that the Web Service is more portable - I can then access the service in whatever language, I just need to create the parser/object transposer in that language.



Because we have WSDL files describing the service, this is almost automated in some systems.



(For example, we have a server written in pure python which is replacing a server written in C, a client written in C++/gSOAP, and a client written in Cocoa/Objective-C. We use soapUI as a testing framework, which is written in Java).

Share Improve this answer

answered Aug 15, 2008 at 2:26

Follow



[Matthew Schinckel](#)

35.6k ● 6 ● 90 ● 121



2

It is possible to return objects from a web service using XML. But Web Services are supposed to be platform and operating system agnostic. Serializing an object simply allows you to store and retrieve an object from a byte stream, such as a file. For instance, you can serialize a



Java object, convert that binary stream (perhaps via a Base 64 encoding into a CDATA field) and transfer that to service's client.



But the client would only be able to restore that object if it were Java-based. Moreover, a deep copy is required to serialize an object and have it restored exactly. Deep copies can be expensive.

Your best route is to create an XML schema that represents the document and create an instance of that schema with the object specifics.

Share Improve this answer

answered Aug 15, 2008 at 3:22

Follow



Brian

771 ● 1 ● 6 ● 18



.NET automatically does this with objects that are serializable. I'm pretty sure Java works the same way.

1

Here is an article that talks about object serialization in

.NET:

http://www.codeguru.com/Csharp/Csharp/cs_syntax/serialization/article.php/c7201



Share Improve this answer

answered Aug 15, 2008 at 2:24

Follow



Seibar

70.2k ● 38 ● 93 ● 100



1



@Brian: I don't know how things work in Java, but in .net objects get serialized down to XML, not base64 strings. The webservice publishes a wsdl file that contains the method and object definitions required for your webservice.

I would hope that nobody creates webservices that simply create a base64 string

Share Improve this answer

answered Aug 15, 2008 at 3:31

Follow



lomaxx

116k ● 58 ● 147 ● 180



1



Daniel Auger:

As others have said, it is possible.

However, if both the service and client use an object that has the exact same domain behavior on both sides, you probably didn't need a service in the first place.

lomax: I have to disagree with this as it's a somewhat narrow comment. Using a webservice that can serialize domain objects to XML means that it makes it easy for clients that work with the same domain objects, but it also means that those clients are restricted to using that particular web service you've exposed and it also works in reverse by allowing other clients to have no

knowledge of your domain objects but still interact with your service via XML.

@ Lomax: You've described two scenarios. **Scenario 1:** The client is rehydrating the xml message back into the exact same domain object. I consider this to be "returning an object". In my experience this is a bad choice and I'll explain this below. **Scenario 2:** The client rehydrates the xml message into something other than the exact same domain object: I am 100% behind this, however I don't consider this to be returning a domain object. It's really sending a message or DTO.

Now let me explain why true/pure/not DTO object serialization across a web service is *usually* a bad idea. An assertion: in order to do this in the first place, you either have to be the owner of both the client and the service, or provide the client with a library to use so that they can rehydrate the object back into it's true type. The problem: This domain object as a type now exists in and belongs to two semi-related domains. Over time, behaviors may need to be added in one domain that make no sense in the other domain and this leads to pollution and potentially painful problems.

I usually default to scenario 2. I only use scenario 1 when there is an overwhelming reason to do so.

I apologize for being so terse with my initial reply. I hope this clears things up to a degree as far as what my opinion is. Lomax, it would seem we half agree ;).

Share Improve this answer

edited Aug 21, 2008 at 15:37

Follow

answered Aug 21, 2008 at 15:31



Daniel Auger

12.6k ● 5 ● 53 ● 73



1



JSON is a pretty standard way to pass objects around the web (as a subset of javascript). Many languages feature a library which will convert JSON code into a native object - see for example [simplejson](#) in Python.

For more libraries for JSON use, see the [JSON webpage](#)



Share Improve this answer

answered Aug 21, 2008 at 16:15

Follow



Brendan

19.3k ● 19 ● 89 ● 114



0



As others have said, it is possible. However, if both the service and client use an object that has the exact same domain behavior on both sides, you probably didn't need a service in the first place.

Share Improve this answer

answered Aug 18, 2008 at 2:21

Follow



Daniel Auger

12.6k ● 5 ● 53 ● 73



0



As others have said, it is possible. However, if both the service and client use an object that has the exact same domain behavior on both sides, you probably didn't need a service in the first place.

I have to disagree with this as it's a somewhat narrow comment. Using a webservice that can serialize domain objects to XML means that it makes it easy for clients that work with the same domain objects, but it also means that those clients are restricted to using that particular web service you've exposed and it also works in reverse by allowing other clients to have no knowledge of your domain objects but still interact with your service via XML.

Share Improve this answer

answered Aug 19, 2008 at 1:43

Follow



lomaxx

116k ● 58 ● 147 ● 180