WCF Service SOAP to REST Converter

# Overview

In some cases, there may be an existing WCF service that uses a standard SOAP binding but that you want to convert to use a REST based endpoint. While this is possible by simply doing some config changes and some additional attributes in the contract, it may be useful to have something that you can just plug in without a recompile.

For example, if you have a contract with an operation defined like this:

[OperationContract]

string BeatTheNiners(int pointSpread);

You can set that up as a RESTful endpoint operation by adding the [WebGet](http://msdn.microsoft.com/en-us/library/system.servicemodel.web.webgetattribute(v=vs.110).aspx) attribute:

[OperationContract]

[WebGet()]

string BeatTheNiners(int pointSpread);

Similarly you can use the [WebInvoke](http://msdn.microsoft.com/en-us/library/system.servicemodel.web.webinvokeattribute(v=vs.110).aspx) attribute if you want a method that’s not GET (POST, PUT, DELETE).

But this requires a recompile. This sample will set up the required behavior without requiring a recompile.

# Setting it up

Put the TheRestifier.dll inside the bin directory of the WCF service.

In the extensions element, declare the RestChangerBehaviorElement:

<system.serviceModel>

<extensions>

<behaviorExtensions>

<add name="RestChanger" type="TheRestifier.Config.RestChangerBehaviorElement, TheRestifier, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />

</behaviorExtensions>

</extensions>

Create the endpoint behavior:

<behaviors>

<endpointBehaviors>

<behavior name="RestBeh">

<RestChanger/>

<webHttp helpEnabled="true" />

</behavior>

</endpointBehaviors>

</behaviors>

Important: make sure that the RestChanger behavior is *before* the webHttp behavior.

Create a new endpoint using this behavior and using the webHttpBinding:

<endpoint name="SeahawksREST"

address="SeahawksREST"

***behaviorConfiguration="RestBeh"***

***binding="webHttpBinding"***

contract="Component.ISeahawks" />

Doing this will set up the endpoint so that all the operations will use the default webGet properties.

If you want to use different methods, uriTemplates or Request/Response formats, then add those properties to the Operation elements:

Example:

<RestChanger>

<Operations>

<Operation name="BeatTheNiners" UriTemplate="BeatTheNiners?PointSpread={pointSpread}" RequestFormat="Json" ResponseFormat="Xml"/>

<Operation name="BeatTheBroncos" method="POST" UriTemplate="BeatTheBroncos?Spread={pointSpread}"/>

</Operations>

</RestChanger>

The default method is GET and the default Request/Response Format is xml.