

LEARNING

LIVES
HERE 

Microsoft®
tech·ed
India | 2009

LEARNING

LIVES
HERE 

Refactoring SOAP/RPC To REST

Niraj Bhatt
Technical Architect
Session Code:



Agenda

- Charge You Up
- Make you start saving money for next Tech Ed
- Deep Dive on migrating a SOAP service to REST service without any change in overall behavior
- Guide you towards in making right decisions (REST vs. SOAP/RPC)

LEARNING

LIVES
HERE 

announcing

Tech Ed

2010

LEARNING

LIVES
HERE 

demo

Nominating Your Speakers For
Tech Ed 2010 – Via a SOAP Service

LEARNING

LIVES
HERE



REST

REST

- 📌 Architectural Style (Resource – Oriented Architecture)
 - Building things the “Web” way
- 📌 YAGNI + KISS (vs. Broad Standardization)
- 📌 Key Principles
 - Uniform Interface (via HTTP Methods)
 - Interoperable (Tech / Platform neutral)
 - Statelessness (Improved Scalability)
 - Caching (Improved Performance)

HTTP Fundamentals

HTTP Methods

GET	PUT	DELETE	POST	HEAD	OPTIONS
-----	-----	--------	------	------	---------

HTTP Status Codes

100	200/201	300/304	400	404	500
-----	---------	---------	-----	-----	-----

LEARNING

LIVES
HERE 

Let's Refactor To REST

Refactoring To REST

Design

- Moving from Verbs to Nouns
- Designing URI Templates
- Selecting Representation Formats

Hosting

Error Handling

Security

Consuming

Moving from Verbs to Nouns

- 📍 Focus on “nouns” (resources) that make up the service
- 📍 Map these nouns to Web’s URI Syntax
- 📍 E.g.
 - speakers
 - speakers/{speakername}
 - speakers?topicname={topicname}

↑
Static Part

↑
Variable Part

Building RESTful Interface

TechEd2010Service

SOAP/RPC	Method	Uri Template
GetAllSpeakers	GET	/speakers
GetAllSpeakersForTopic	GET	/speakers?topic={topic}
GetSpeaker	GET	/speakers/{speakername}
NominateSpeaker	PUT (POST)	/speakers/{speakername}
DeleteSpeaker	DELETE	/speakers/{speakername}
UpdateSpeaker	PUT	/speakers/{speakername} /{topicname}

Representation Formats

POX

- Supported by all major frameworks

JSON

- AJAX Web enabled clients

ATOMPub

- Simple HTTP-based protocol for creating and updating web resources (Atom 1.0 Feed and Entry documents)

Supporting POX & JSON

```
[OperationContract]  
[HttpGet(UriTemplate = "speakers")]  
Speakers GetAllSpeakers();
```



```
[OperationContract]  
[HttpGet(UriTemplate = "speakers?format=json",  
         ResponseFormat = WebMessageFormat.Json)]  
Speakers GetAllSpeakersAsJSON();
```

WCF Implementation

```
S binding="webHttpBinding" behaviorConfiguration="enableWebHttp"  
T <endpointBehaviors>  
E     <behavior name="enableWebHttp">  
P         <webHttp />  
1     </behavior>  
    </endpointBehaviors>
```

```
S [OperationContract]  
T [WebGet(UriTemplate = "speakers?topicName={topicName}")]  
E List<Speaker> GetAllSpeakersForTopic(string topicName);  
P [OperationContract]  
2 [WebInvoke(Method = "PUT", UriTemplate = "speakers/{speakerName}")]  
   void NominateSpeaker(Speaker speaker);
```


Hi-REST vs. Lo-REST

Thinking Higher...

- Appropriate Verbs
- Representation Formats
- Appropriate Response Codes
- URI Design
- Etc...



LEARNING

LIVES
HERE 

demo

Testing RESTful Service via Fiddler

REST Starter Kit



I'm not sure I want to build on a layer designed to factor HTTP in, on top of a layer that was designed to factor it out.
– Tim Ewald



REST Starter Kit

REST Starter Kit Contents

WebServiceHost2Factory

WebServiceHost2

VS.NET Templates

PollingAgent

WebProtocolException

RequestInterceptor

Paste XML as Types

HttpResponseMessage

HttpClient

AdapterStream

LEARNING

LIVES
HERE 

demo

Using WCF REST Starter Kit – Hosting &
Error Handling

Security – Basic Authentication + SSL

- IIS Basic Authentication
 - Limited to Windows Credentials
- RequestInterceptor
 - 401 – Unauthorized
 - WWW-Authenticate Header
 - SSL (Default Encoding Base 64)



LEARNING

LIVES
HERE 

demo

Securing RESTful Service using Basic
Authentication over HTTPS

Consuming RESTful Service

~~//Add Web Reference / Add Service Reference~~

```
HttpClient client = new HttpClient();
HttpResponseMessage message = client.Get("Your Uri..");
//Put, Delete, Post
using(message)
{
    if (message.StatusCode == HttpStatusCode.OK)
    {
        //...
    }
}

//Paste XML as Types
```

Summary

- Refactoring SOAP to REST is easy, but requires a paradigm shift
- The best decisions come from reason, not emotion
- Questions you can ask
 - Does the service expose data or logic?
 - Does the service need the capabilities of WS-*?
 - What's best for the consuming developers?

ଆଭାର

ଧନ୍ୟବାଦ

நன்றி

धन्यवाद

Thank You!

ధన్యవాదాలు

ಧನ್ಯವಾದಗಳು

ଧନ୍ୟବାଦ

ಧನ್ಯವಾದಗಳು

ਧੰਨਵਾਦ

നീങ്ങളുടെ നന്മ

LEARNING

LIVES
HERE 

Q & A



What's NEXT

Blog: <http://nirajrules.wordpress.com> Email: niraj@indiamvps.net

REST on MSDN: <http://msdn.microsoft.com/en-us/netframework/cc950529.aspx>

Blogs: David Chappell, Aaron Skonnard, Ron Jacobs, Rob Bagby, Pablo Cibraro

WCF REST Starter Kit:

<http://aspnet.codeplex.com/Release/ProjectReleases.aspx?ReleaseId=24644>

WCF REST Starter Kit Hands On Labs:

<http://code.msdn.microsoft.com/wcfrestlabs/Release/ProjectReleases.aspx?ReleaseId=2368>

Related Content

ADO.NET Data Services Framework – Harish Ranganathan

Creating REST Enterprise Mashups using SharePoint Designer 2007 – Abhishek Banerjee

Federated Identity Architectures: Integrating with the Cloud – Ramnish Singh, Niraj Bhatt

WCF 3.5: Unit Testing Windows Communication Foundation 3.5 - Rajapadmanaban Narayanaswamy

Load Testing within Team System - Rajapadmanaban Narayanaswamy

Microsoft[®]

Your potential. Our passion.[™]

© 2009 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.