REST

Silesian Ruby Users' Group
Szymon Nowak

April 27, 2009

Web services

Web services

what exactly are web services?

Web services

API for web applications

Web services

some examples:

- weather
- sport results
- stock market

Web services

a bit of history

Web services

Remote Procedure Call

Web services

XML-RPC

Web services

uses XML over HTTP

XML RPC sample request

```
<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
        <value><i4>40</i4></value>
    </param>
  </params>
</methodCall>
```

XML RPC sample response

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
        <value><string>South Dakota</string></value>
    </param>
  </params>
</methodResponse>
```

Web services

this evolved into

Web services

SOAP

Web services

Simple Object Access Protocol

Web services

this acronym was dropped with version 1.2 of the standard

- it was confused with SOA
- it's not that simple after all

Web services

uses XML over HTTP

SOAP sample request

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetEndorsingBoarder xmlns:m="http://namespaces.snowboard-info.com">
      <manufacturer>K2</manufacturer>
      <model>Fatbob</model>
    </m:GetEndorsingBoarder>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP sample response

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetEndorsingBoarderResponse xmlns:m="http://namespaces.snowboard-info.co
      <endorsingBoarder>Chris Englesmann</endorsingBoarder>
    </m:GetEndorsingBoarderResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Web services

services are defined using
Web Services Description Language
(WSDL)

WSDL sample

```
<?xml version="1.0"?>

<!-- root element wsdl:definitions defines set of related services -->
<wsdl:definitions name="EndorsementSearch"
  targetNamespace="http://namespaces.snowboard-info.com"
  xmlns:es="http://www.snowboard-info.com/EndorsementSearch.wsdl"
  xmlns:esxsd="http://schemas.snowboard-info.com/EndorsementSearch.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

  <!-- wsdl:types encapsulates schema definitions of communication types; here using xsd -->
  <wsdl:types>

    <!-- all type declarations are in a chunk of xsd -->
    <xsd:schema targetNamespace="http://namespaces.snowboard-info.com"
      xmlns:xsd="http://www.w3.org/1999/XMLSchema">

      <!-- xsd definition: GetEndorsingBoarder [manufacturer string, model string] -->
      <xsd:element name="GetEndorsingBoarder">
        <xsd:complexType>
            <xsd:sequence>
                  <xsd:element name="manufacturer" type="string"/>
            <xsd:element name="model" type="string"/>
            </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
```

Web services

... about 100 lines of XML later ...

WSDL sample

```
  <!-- wsdl:service names a new service "EndorsementSearchService" -->
  <wsdl:service name="EndorsementSearchService">
    <wsdl:documentation>snowboarding-info.com Endorsement Service</wsdl:documentation>

    <!-- connect it to the binding "EndorsementSearchSoapBinding" above -->
    <wsdl:port name="GetEndorsingBoarderPort"
               binding="es:EndorsementSearchSoapBinding">

      <!-- give the binding an network address -->
      <soap:address location="http://www.snowboard-info.com/EndorsementSearch"/>
    </wsdl:port>
  </wsdl:service>

</wsdl:definitions>
```

Web services

- lets tools create client APIs
- client developers see methods with parameters

Web services

WS-* specifications

- WS-Addressing
- WS-Security
- WS-Trust
- WS-SecureConversation
- WS-ReliableMessaging
- WS-AtomicTransaction
- WS-Coordination
- WS-Policy
- WS-MetadataExchange
- ...

Web Services Standards Overview

Interoperability Issues

Basic Profile 1.1 · WS-I · Final Specification
▲ **Basic Profile** – The Basic Profile 1.1 provides implementation guidelines for how related set of non-proprietary Web Service specifications should be used together for best interoperability.

Basic Profile 1.2 · WS-I · Working Group Draft
▲ **Basic Profile** – The Basic Profile 1.2 builds on Basic Profile 1.1 by incorporating Basic Profile 1.1 errata, requirements from Simple SOAP Binding Profile 1.0, and adding support for WS-Addressing and MTOM.

Basic Profile 2.0 · WS-I · Working Group Draft
▲ **Basic Profile** – The Basic Profile 2.0 is an update of WS-I BP that includes a profile of SOAP 1.2.

Attachments Profile 1.0 · WS-I · Final Specification
▲ **Attachments Profile** – The Attachment Profile 1.0 complements the Basic Profile 1.1 to add support for interoperable SOAP Messages with attachments-based Web Services.

Simple SOAP Binding Profile 1.0 · WS-I · Final Specification
▲ **Simple SOAP Binding Profile** – The Simple SOAP Binding Profile consists of those Basic Profile 1.0 requirements related to the serialization of the envelope and its representation in the message.

Basic Security Profile 1.0 · WS-I · Board Approval Draft
▲ **Basic Security Profile** defines the WS-I Basic Security Profile 1.0, based on a set of non-proprietary Web services specifications, along with clarifications and amendments to those specifications which promote interoperability.

REL Token Profile 1.0 · WS-I · Working Group Draft
▲ **REL Token Profile** is based on a non-proprietary Web services specification, along with clarifications and amendments to that specification which promote interoperability.

SAML Token Profile 1.0 · WS-I · Working Group Draft
▲ **SAML Token Profile** is based on a non-proprietary Web services specification, along with clarifications and amendments to that specification which promote interoperability.

Conformance Claim Attachment Mechanism (CCAM) 1.0 · WS-I · Final Specification
▲ **Conformance Claim Attachment Mechanism (CCAM)** catalogues mechanisms that can be used to attach WS-I Profile Conformance Claims to Web services artefacts (e.g., WSDL descriptions, UDDI registries).

Reliable Asynchronous Messaging Profile (RAMP) 1.0 · WS-I · Working Draft
▲ **Reliable Asynchronous Messaging Profile (RAMP)** is a profile, in the fashion of the WS-I profiles, that enables, among other things, basic B2B integration scenarios using Web services technologies.

Standards Bodies

OASIS – The **Organization for the Advancement of Structured Information Standards (OASIS)** is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. The consortium produces more Web services standards than any other organization along with standards for security, e-business, and standardization efforts in the public sector and for application-specific markets. Founded in 1993, OASIS has more than 4,000 participants representing over 600 organizations and individual members in 100 countries.

W3C – The **World Wide Web Consortium (W3C)** was created in October 1994 to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. W3C has over 350 Member organizations from all over the world and has earned international recognition for its contributions to the growth of the Web. W3C is designing the infrastructure, and defining the architecture and the core technologies for Web services. In September 2000, W3C started the XML Protocol Activity to address the need for an XML-based protocol for application-to-application messaging. In January 2002, the Web Services Activity was launched, subsuming the XML Protocol Activity and extending its scope.

WS-I – The **Web Services Interoperability Organization (WS-I)** is an open industry organization chartered to promote Web services interoperability across platforms, operating systems and programming languages. The organization's diverse community of Web services leaders helps customers to develop interoperable Web services by providing guidance, recommended practices and supporting resources. Specifically, WS-I creates, promotes and supports generic protocols for the interoperable exchange of messages between Web services.

IETF – The **Internet Engineering Task Force (IETF)** is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

Business Process Specifications

Business Process Execution Language for Web Services 1.1 (BPEL4WS) · 1.1 · BEA Systems, IBM, Microsoft, SAP, Siebel Systems · OASIS-Standard
▲ **Business Process Execution Language for Web Services 1.1** (BPEL4WS) provides a language for the formal specification of business processes and business interaction protocols using Web Services.

WS-Choreography Model Overview 1.0 · W3C · Working Draft
▲ **WS-Choreography Model Overview** defines the format and structure of the (SOAP) messages that are exchanged, and the sequence and conditions in which the messages are exchanged.

Web Service Choreography Interface (WSCI) · 1.0 · W3C · Sun Microsystems, SAP, BEA Systems and Intalio · Note
▲ **Web Service Choreography Interface** (WSCI) describes how Web Service operations can be choreographed in the context of a message exchange in which the Web Service participates.

Web Service Choreography Description Language (CDL4WS) · 1.0 · W3C · Candidate Recommendation
▲ **Web Service Choreography Description Language** (CDL4WS) is to specify a declarative, XML based language that defines from a global viewpoint the common and complementary observable behaviour, where message exchanges occur, and when the jointly agreed ordering rules are satisfied.

Business Process Execution Language for Web Services 2.0 (BPEL4WS) · 2.0 · OASIS, BEA Systems, IBM, Microsoft, SAP, Siebel Systems · Committee Draft
▲ **Business Process Execution Language for Web Services 2.0** (BPEL4WS) provides a language for the formal specification of business processes and business interaction protocols using Web Services.

Business Process Management Language (BPML) 1.1 · BPMI.org · Final Draft
▲ **Business Process Management Language** (BPML) provides a meta-language for expressing business processes and supporting entities.

XML Process Definition Language (XPDL) 2.0 · Final
▲ **XML Process Definition Language** (XPDL) provides an XML file format that can be used to interchange process models between tools.

Management Specifications

Management Using Web Services (WSDM-MUWS) 1.0 · OASIS · OASIS-Standard
▲ **Web Service Distributed Management:** Management Using Web Services (WSDM-MUWS) defines how an IT resource connected to a network provides manageability interfaces such that the IT resource can be managed locally and from remote locations using Web services technologies.

Management Of Web Services (WSDM-MOWS) 1.0 · OASIS · OASIS-Standard
▲ **Web Service Distributed Management:** Management Of Web Services (WSDM-MOWS) addresses management of the components that form the network, the Web services endpoints, using Web services protocols.

WS-Management AMD, Dell, Intel, Microsoft and Sun Microsystems · Published Specification
▲ **WS-Management** describes a general SOAP-based protocol for managing systems such as PCs, servers, devices, Web services and other applications, and other manageable entities.

Service Modeling Language IBM, BEA, BMC, Cisco, Dell, HP, Intel, Microsoft, Sun · Draft Specification
▲ **Service Modeling Language** (SML) is used to model complex IT services and systems, including their structure, constraints, policies, and best practices.

Presentation Specifications

Web Services for Remote Portlets (WSRP) 2.0 · OASIS · Committee Draft
▶ **Web Services for Remote Portlets** (WSRP) defines a set of interfaces and related semantics which standardize interactions with components providing user-facing markup, including the processing of user interactions with that markup.

Metadata Specifications

WS-Policy 1.5 · W3C · Working Draft
▲ **WS-Policy** describes the capabilities and constraints of the policies on intermediaries and endpoints (e.g. business rules, required security tokens, supported encryption algorithms, privacy rules).

WS-PolicyAssertions 1.1 · BEA Systems, IBM, Microsoft, SAP · Public Draft
▲ **WS-PolicyAssertions** provides an initial set of assertions to address some common needs of Web Services applications.

WS-PolicyAttachment 1.2 · W3C · W3C Member Submission
▲ **WS-PolicyAttachment** defines two general-purpose mechanisms for associating policies with the subjects to which they apply; the policies may be defined as part of existing metadata about the subject or the policies may be defined independently and associated through an external binding to the subject.

WS-Discovery Microsoft, BEA Systems, Canon, Intel and webMethods · Draft
▲ **WS-Discovery** defines a multicast discovery protocol for dynamic discovery of services on ad-hoc and managed networks.

WS-MetadataExchange 1.1 · BEA Systems, Computer Associates, IBM, Microsoft, SAP, Sun Microsystems and webMethods · Public Draft
▲ **WS-MetadataExchange** enables a service to provide metadata to others through a Web services interface. Given only a reference to a Web service, a user can access a set of WSDL/SOAP operations to retrieve the metadata that describes the service.

Universal Description, Discovery and Integration (UDDI) 3.0.2 · OASIS · OASIS-Standard
▲ **Universal Description, Discovery and Integration (UDDI)** defines a set of services supporting the description and discovery of businesses, organizations, and other Web services providers, the Web services they make available, and the technical interfaces which may be used to access those services.

Web Service Description Language 2.0 SOAP Binding 2.0 · W3C · Working Draft
▲ **Web Service Description Language SOAP Binding** describes the concrete details for using WSDL 2.0 in conjunction with SOAP 1.1 protocol.

Web Service Description Language 2.0 Core 2.0 · W3C · Candidate Recommendation
▲ **Web Service Description Language 2.0 Core** is an XML-based language for describing Web services and how to access them. It specifies the location of the service and the operations (or methods) the service exposes.

Web Service Description Language 1.1 1.1 · W3C · Note
▲ **Web Service Description Language 1.1** is an XML-based language for describing Web services and how to access them. It specifies the location of the service and the operations (or methods) the service exposes.

Reliability Specifications

WS-ReliableMessaging 1.1 · OASIS · Committee Draft
▲ **WS-ReliableMessaging** describes a protocol that allows Web services to communicate reliable in the presence of software component, system, or network failures. It defines a SOAP binding that is required for interoperability.

WS-Reliable Messaging Policy Assertion (WS-RM Policy) 1.1 · OASIS · Committee Draft
▲ **Web Services ReliableMessaging Policy Assertion** (WS-RM Policy) describes a domain-specific policy assertion for WS-ReliableMessaging that that can be specified within a policy alternative as defined in WS-Policy Framework.

WS-Reliability 1.1 · OASIS · OASIS-Standard
▲ **WS-Reliability** is a SOAP-based protocol for exchanging SOAP messages with guaranteed delivery, no duplicates, and guaranteed message ordering. WS-Reliability is defined as SOAP header extensions and is independent of the underlying protocol. This specification contains a binding to HTTP.

Security Specifications

WS-Security 1.1 · OASIS · OASIS-Standard
▲ **WS-Security** is a communications protocol providing a means for applying security to Web Services.

WS-SecurityPolicy 1.1 · IBM, Microsoft, RSA Security, VeriSign · Public Draft
▲ **WS-SecurityPolicy** defines how to describe policies related to various features defined in the WS-Security specification.

WS-Security: SOAP Message Security 1.1 · OASIS · Public Review Draft
▲ **WS-Security: SOAP Message Security** describes enhancements to SOAP messaging to provide message integrity and confidentiality. Specifically, this specification provides support for multiple security token formats, trust domains, signature formats and encryption technologies. The token formats and semantics for using these are defined in the associated profile documents.

WS-Security: Username Token Profile 1.1 · OASIS · Public Review Draft
▲ **WS-Security: Username Token Profile** describes how a Web Service consumer can supply a Username Token as a means of identifying the requestor by username, and optionally using a password (or shared secret, etc.) to authenticate that identity to the Web Service producer.

WS-Security: Kerberos Binding 1.0 · Microsoft, IBM, OASIS · Working Draft
▲ **WS-Security: Kerberos Binding** defines how to encode Kerberos tickets and attach them to SOAP messages. As well, it specifies how to add signatures and encryption to the SOAP message, in accordance with WS-Security, which uses and references the Kerberos tokens.

WS-Federation 1.0 · IBM, Microsoft, BEA Systems, RSA Security, and VeriSign · Initial Draft
▲ **WS-Federation** describes how to manage and broker the trust relationships in a heterogeneous federated environment including support for federated identities.

WS-Security: SAML Token Profile 1.1 · OASIS · Public Review Draft
▲ **WS-Security: SAML Token Profile** defines the use of Security Assertion Markup Language (SAML) v1.1 assertions in the context of WSS: SOAP Message Security including for the purpose of securing SOAP messages and SOAP message exchanges.

WS-Trust BEA Systems, Computer Associates, IBM, Layer 7 Technologies, Microsoft, Netegrity, Oblix, OpenNetwork, Ping Identity Corporation, Reactivity, RSA Security, VeriSign and Westbridge Technology · Initial Draft
▲ **WS-Trust** describes a framework for trust models that enables Web Services to securely interoperate. It uses WS-Security base mechanisms and defines additional primitives and extensions for security token exchange to enable the issuance and dissemination of credentials within different trust domains.

WS-Security: X.509 Certificate Token Profile 1.1 · OASIS · Public Review Draft
▲ **WS-Security: X.509 Certificate Token Profile** describes the use of the X.509 authentication framework with the WS-Security: SOAP Message Security specification.

WS-SecureConversation BEA Systems, Computer Associates, IBM, Layer 7 Technologies, Microsoft, Netegrity, Oblix, OpenNetwork, Ping Identity Corporation, Reactivity, RSA Security, VeriSign and Westbridge Technology · Public Draft
▲ **WS-SecureConversation** specifies how to manage and authenticate message exchanges between parties including security context exchange and establishing and deriving session keys.

Transaction Specifications

WS-Coordination 1.1 · OASIS · Working Draft
▲ **WS-Coordination** describes an extensible framework for providing protocols that coordinate the actions of distributed applications.

WS-Business Activity 1.1 · OASIS · Working Draft
▲ **WS-Business Activity** provides the definition of the business activity coordination type that is to be used with the extensible coordination framework described in the WS-Coordination specification.

WS-Atomic Transaction 1.1 · OASIS · Committee Draft
▲ **WS-Atomic Transaction** defines protocols that enable existing transaction processing systems to wrap their proprietary protocols and interoperate across different hardware and software vendors.

WS-Composite Application Framework (WS-CAF) 1.0 · Arjuna Technologies, Fujitsu, IONA, Oracle and Sun Microsystems · Committee Specification
▲ **WS-Composite Application Framework** (WS-CAF) is a collection of three specifications aimed at solving problems that arise when multiple Web Services are used in combination. It proposes standard, interoperable mechanisms for managing shared context and ensuring business processes achieve predictable results and recovery from failure.

WS-Context (WS-CTX) 1.0 · Arjuna Technologies, Fujitsu, IONA, Oracle and Sun Microsystems · Committee Draft
▲ **WS-Context** (WS-CTX) is intended as a lightweight mechanism for allowing multiple Web Services to share a common context.

WS-Coordination Framework (WS-CF) 1.0 · Arjuna Technologies, Fujitsu, IONA, Oracle and Sun Microsystems · Committee Draft
▲ **WS-Coordination Framework** (WS-CF) allows the management and coordination in a Web Services interaction of a number of activities related to an overall application.

WS-Transaction Management (WS-TXM) 1.0 · Arjuna Technologies, Fujitsu, IONA, Oracle and Sun Microsystems · Committee Draft
▲ **WS-Transaction Management** (WS-TXM) defines a core infrastructure service consisting of a Transaction Service for Web Services.

Resource Specifications

Web Services Resource Framework (WSRF) 1.2 · OASIS · OASIS-Standard
▲ **Web Services Resource Framework** (WSRF) defines a family of specifications for accessing stateful resources using Web Services.

WS-BaseFaults (WSRF) 1.2 · OASIS · Working Draft
▲ **WS-BaseFaults** (WSRF) defines a base set of information that may appear in fault messages. WS-BaseFaults defines an XML schema type for base faults, along with rules for how this base fault type is used and extended by Web Services.

WS-ServiceGroup (WSRF) 1.2 · OASIS · Working Draft
▲ **WS-ServiceGroup** (WSRF) defines a means by which Web Services and WS-Resources can be aggregated or grouped together for a domain specific purpose.

WS-ResourceProperties 1.2 · OASIS · Working Draft
▲ **WS-ResourceProperties** specifies the means by which the definition of the properties of a WS-Resource may be declared as part of the Web Service interface. The declaration of the WS-Resource properties represents a projection of or a view on the WS-Resource state.

WS-ResourceLifetime 1.2 · OASIS · Working Draft
▲ **WS-ResourceLifetime** is to standardize the terminology, concepts, message exchanges, WSDL and XML needed to monitor the lifetime of, and destroy WS-Resources. Additionally, it defines resource properties that may be used to inspect and monitor the lifetime of a WS-Resource.

WS-Transfer W3C · W3C Member Submission
▲ **WS-Transfer** describes a general SOAP-based protocol for accessing XML representations of Web service-based resources.

Resource Representation SOAP Header Block (RRSHB) W3C · Recommendation
▲ **Resource Representation SOAP Header Block** (RRSHB) complements MTOM by defining mechanisms for describing and conveying non-XML resource representations in a SOAP 1.2 message.

Messaging Specifications

WS-Notification 1.3 · OASIS · OASIS-Standard
▶ **WS-Notification** is a family of related white papers and specifications that define a standard Web services approach to notification using a topic-based publish/subscribe pattern.

WS-BrokeredNotification 1.3 · OASIS · OASIS-Standard
▶ **WS-BrokeredNotification** defines the interface for the NotificationBroker. A NotificationBroker is an intermediary, which, among other things, allows publication of messages from entities that are not themselves service providers.

WS-BaseNotification 1.3 · OASIS · OASIS-Standard
▶ **WS-BaseNotification** standardizes the terminology, concepts, operations, WSDL and XML needed to express the basic roles involved in Web services publish and subscribe for notification message exchange.

WS-Eventing W3C · Public Draft
▶ **WS-Eventing** defines a baseline set of operations that allow Web services to provide asynchronous notifications to interested parties.

WS-Addressing – Core 1.0 · W3C · Recommendation
▶ **WS-Addressing – Core** provides transport-neutral mechanisms to address Web services and messages. This specification defines XML elements to identify Web service endpoints and to secure end-to-end endpoint identification in messages.

WS-Enumeration Systinet, Microsoft, Sonic Software, BEA Systems and Computer Associates · Public Draft
▶ **WS-Enumeration** describes a general SOAP-based protocol for enumerating a sequence of XML elements that is suitable for traversing logs, message queues, or other linear information models.

WS-Topics 1.3 · OASIS · OASIS-Standard
▶ **WS-Topics** defines three topic expression dialects that can be used as subscription expressions in subscribe request messages and other parts of the WS-Notification system.

WS-Addressing – WSDL Binding 1.0 · W3C · Candidate Recommendation
▶ **WS-Addressing – WSDL Binding** defines how the abstract properties defined in Web Services Addressing – Core are described using WSDL.

WS-Addressing – SOAP Binding 1.0 · W3C · Recommendation
▶ **WS-Addressing – SOAP Binding** provides transport-neutral mechanisms to address Web services and messages.

SOAP

SOAP 1.2 · W3C · Recommendation

SOAP Message Transmission Optimization Mechanism (MTOM) 1.0 · W3C · Recommendation
▶ **SOAP Message Transmission Optimization Mechanism** describes an abstract feature for optimizing the transmission and/or wire format of a SOAP message.

SOAP 1.1 · W3C · Note
▶ **SOAP** is a lightweight, XML-based protocol for exchange of information in a decentralized, distributed environment.

XML Specifications

XML 1.1 1.1 · W3C · Recommendation
▶ **XML** – Extensible Markup Language is a pared-down version of SGML, designed especially for Web documents. It allows one to create own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

XML 1.0 1.0 · W3C · Recommendation
▶ **XML** – Extensible Markup Language is a pared-down version of SGML, designed especially for Web documents. It allows one to create own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

Namespaces in XML 1.1 · W3C · Recommendation
▶ **Namespaces in XML** provides a simple method for qualifying element and attribute names used in XML documents by associating them with namespaces identified by IRI references.

XML Information Set 1.0 · W3C · Recommendation
▶ **XML Information Set** is an abstract data set to provide a consistent set of definitions for use in other specifications that need to refer to the information in a well-formed XML document.

XML Schema 1.1 · W3C · Working Draft
▶ **XML Schema** – XML Schema Definition Language is an XML language for describing and constraining the content of XML documents.

XML binary Optimized Packaging (XOP) 1.0 · W3C · Recommendation
▶ **XML binary Optimized Packaging (XOP)** is an XML language for describing and constraining the content of XML documents.

Describing Media Content of Binary Data in XML (DMCBDX) · W3C · Note
▶ **Describing Media Content of Binary Data in XML (DMCBDX)** specifies how to indicate the content-type associated with binary element content in an XML document and to specify, in XML Schema, the expected content-type(s) associated with binary element content.

Dependencies

Messaging Specifications
- SOAP 1.1
- SOAP 1.2
- SOAP Message Transmission Optimization Mechanism
- WS-Notification
- WS-BaseNotification
- WS-Topics
- WS-BrokeredNotification
- WS-Addressing – Core
- WS-Addressing – WSDL Binding
- WS-Addressing – SOAP Binding
- WS-Eventing
- WS-Enumeration

Resource · Security · Metadata

Metadata Specifications
- WS-Policy
- WS-PolicyAssertions
- WS-PolicyAttachment
- WS-Discovery
- WS-MetadataExchange
- Universal Description, Discovery and Integration
- Web Service Description Language 1.1
- Web Service Description Language 2.0 Core
- Web Service Description Language 2.0 SOAP Binding

Security · Messaging

Security Specifications
- WS-Security
- WS-Security: SOAP Message Security
- WS-Security: Kerberos Binding
- WS-Security: SAML Token Profile
- WS-Security: X.509 Certificate Token Profile
- WS-Security: Username Token Profile
- WS-SecurityPolicy
- WS-Trust
- WS-Federation
- WS-SecureConversation

Reliability · Messaging · Metadata

Reliability Specifications
- WS-ReliableMessaging
- WS-Reliability
- WS-Reliable Messaging Policy Assertion

Transaction · Basic Profile · Security · Metadata

Resource Specifications
- Web Service Resource Framework
- WS-BaseFaults
- WS-ServiceGroup
- WS-ResourceProperties
- WS-ResourceLifetime
- WS-Transfer
- Resource Representation SOAP Header Block (RRSHB)

Transaction · Security · Messaging

Management Specifications
- WS-Management
- Management Of Web Services
- Management Using Web Services
- Service Modeling Language

Meta · Resource · Security · Messaging

Business Process Specifications
- Business Process Execution Language for Web Services
- Web Service Choreography Description Language
- Web Service Choreography Interface
- WS-Choreography Model Overview
- Business Process Management Language
- Business Process Execution Language for Web Serv. 2.0
- XML Process Definition Language

Reliability · Security · Messaging · Transaction

Transaction Specifications
- WS-Business Activity
- WS-Atomic Transaction
- WS-Coordination
- WS-Composite Application Framework
- WS-Transaction Management
- WS-Context
- WS-Coordination Framework

Metadata · Messaging · Reliability · Security

Presentation Specifications
- Web Services for Remote Portlets

Reliab. · Secur. · Mess.

innoQ

innoQ Deutschland GmbH
Halskestraße 17
D-40880 Ratingen
Phone +49 21 02 77 162-100
info@innoq.com · www.innoq.com

innoQ Schweiz GmbH
Gewerbestrasse 11
CH-6330 Cham
Phone +41 41 743 01 11

Version 3.0 · February 2007

Web services

service oriented design

Web services

- UserManager
    - createUser(u:User)
    - getUserDetails(id:ID)
- StatusManager
    - submitStatus(u_id:ID, s:Status)
    - getStatus(u_id:ID)

Web services

Cons of SOAP services:

- complex
- strong typing
- XML is not necessarily the best data format for the web
- non-uniform interface
- uses HTTP POST only

Web services

not everyone needs enterprisey and complex web services

Web services

you don’t have to use SOAP

Web services

others don’t

Dirty Harry

Web services

- Amazon Web Services - provides both
  - 20% uses SOAP
  - 80% uses REST
- Google Search API - deprecated SOAP in favor of REST
- Yahoo API - uses REST only

REST

REST

REST

REpresentational State Transfer

REST

introduced by Roy Fielding, who also worked on the following specifications:

- URI
- HTTP
- HTML

REST

very short demo

REST

URI

REST

resources

REST

uniquely addressable using URIs

REST

http://localhost/users/1

REST

http://localhost/users/1/statuses/1

REST

http://localhost/users

REST

http://localhost/users/1/statuses

REST

HTTP

REST

CRUD

REST

ACTION

CREATE
READ
UPDATE
DELETE

REST

ACTION	SQL
CREATE	INSERT
READ	SELECT
UPDATE	UPDATE
DELETE	DELETE

REST

ACTION	SQL	HTTP
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT
DELETE	DELETE	DELETE

REST

ACTION	SQL	HTTP
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT
DELETE	DELETE	DELETE

REST

ACTION	SQL	HTTP
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT
DELETE	DELETE	DELETE

REST

ACTION	SQL	HTTP
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT
DELETE	DELETE	DELETE

REST

ACTION	SQL	HTTP
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT
DELETE	DELETE	DELETE

REST

think of REST as a sentence:

- HTTP actions are verbs
- resources' URIs are nouns

REST

POST	http://localhost/users
GET	http://localhost/users/1
PUT	http://localhost/users/1
DELETE	http://localhost/users/1

REST

POST	http://localhost/users
GET	http://localhost/users/1
PUT	http://localhost/users/1
DELETE	http://localhost/users/1

REST

uniform interface to interact with resources

REST

POST	http://localhost/users/1/statuses
GET	http://localhost/users/1/statuses/1
PUT	http://localhost/users/1/statuses/1
DELETE	http://localhost/users/1/statuses/1

REST

resources can have many representations

REST

“Get XML representation of user with ID 1”

```
GET     http://localhost/users/1.xml
GET     Accept: application/xml http://localhost/users/1
```

REST

“Get JSON representation of user with ID 1”

GET	http://localhost/users/1.json
GET	Accept: application/json http://localhost/users/1

REST

“Get HTML representation of user with ID 1”

```
GET     http://localhost/users/1.html
GET     Accept: text/html http://localhost/users/1
```

REST

"Get vCard representation of user with ID 1"

```
GET    http://localhost/users/1.vcf
GET    Accept: text/x-vCard http://localhost/users/1
```

REST

REST is not a standard
it's a style of software architecture

REST in Rails

REST in Rails

in Rails it's easier to build RESTful
than non-RESTful apps

REST in Rails

quick demo

REST in Rails

how does it work?

REST in Rails

REST actions

POST
GET
PUT
DELETE

REST in Rails

Rails actions

create
show
update
destroy
new
edit
index

REST in Rails

Rails actions

create
show
update
destroy
new
edit
index

REST in Rails

7 default actions

REST in Rails

Rails actions	HTTP request	
create	POST	/users
show	GET	/users/1
update	PUT	/users/1
destroy	DELETE	/users/1
index	GET	/users
new	GET	/users/1/new
edit	GET	/users/1/edit

REST in Rails

how does Rails know how to map URI to an action?

REST in Rails

routes

config/routes.rb

```ruby
ActionController::Routing::Routes.draw do |map|
  map.resources :users
end
```

REST in Rails

generates mapping for 7 default actions
for user resource

REST in Rails

generates helper methods for 7 default actions
for user resource

REST in Rails

Rails actions	URI	helpers
create	/users	users_path
show	/users/1	user_path(1)
update	/users/1	user_path(1)
destroy	/users/1	user_path(1)
index	/users	users_path
new	/users/1/new	new_user_path
edit	/users/1/edit	edit_user_path(1)

REST in Rails

resource representations

REST in Rails

respond_to

app/controllers/users_controller.rb

```ruby
class UsersController < ApplicationController

  # GET /users/1
  # GET /users/1.xml
  def show
    @user = User.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.xml  { render :xml => @user }
    end
  end

end
```

Consuming RESTful web services

Consuming RESTful web services

Consuming RESTful web services

system tools

- cURL

Consuming RESTful web services

we get raw XML/JSON response
that we still need to parse

FAIL
failblog.org

Consuming RESTful web services

Ruby libraries

- HTTParty
- ActiveResource

Consuming RESTful web services

HTTParty

Consuming RESTful web services

for RESTful and RESTful-like web services

Consuming RESTful web services

what does it do for you:

- sends request
- processes response

HTTParty client example

```
class TwitterCloneClient
  include HTTParty
  base_uri "localhost:3000"
  format :xml
end

TwitterCloneClient.get("/statuses/1")
#{"status"=>{
#  "id"=>1,
#  "body"=>"First message",
#  "created_at"=>Wed Apr 26 20:38:19 UTC 2009,
#  "user_id"=>1...}
#}
```

Consuming RESTful web services

demo

Consuming RESTful web services

ActiveResource

Consuming RESTful web services

for strictly RESTful web services

Consuming RESTful web services

- part of Rails core
- works best with Rails apps
- provides ActiveRecord like API to RESTful web services

Consuming RESTful web services

what does it do for you:

- forms request URI
- sends request
- processes response
- provides OO access to response

Consuming RESTful web services

how to write a client?

ActiveResource client example

```ruby
class Status < ActiveResource::Base
  self.site = "http://localhost:3000/"
end

# Find
status = Status.find(:first)
status.body # => "First message"
```

ActiveResource client example

```
# Create
status = Status.create(:body => "New messsage")

# Update
status.body = "Updated"
status.save

# Delete
status.destroy
```

Consuming RESTful web services

demo