

CS 683 Emerging Technologies
Fall Semester, 2006
Doc 26 Rails, XML-RPC & SOAP
Nov 30, 2006

Copyright ©, All rights reserved. 2006 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

Agile Web Development with Rails 2nd Ed Bl.16 October 25, Thomas & Hanson, The Pragmatic Bookshelf, PDF

How to Deal with Remote Objects?

Server classes

```
XmlRpcClient xmlrpc = new XmlRpcClientLite  
    ("http://xmlrpc.usefulinc.com/demo/server.php");  
Vector parameters = new Vector ();  
parameters.addElement ( "College Area" );  
Restaurant x =  
    (Restaurant) xmlrpc.execute ("restaurant.findFirst", parameters);
```

Our Restaurant class is implemented in Ruby not Java?

1. Make Each Client implement Class

```
class Restaurant {  
    etc.  
}
```

```
Restaurant x = (Restaurant) xmlrpc.execute ("restaurant.findFirst", parameters);
```

Implementing client requires re-implementing server classes

2. Pass Remote Proxy

```
Restaurant x = (Restaurant) xmlrpc.execute ("restaurant.findFirst", parameters);  
x.name()    /*remote method call */
```

Server passes remote proxy to client

Every method sent to proxy is forwarded back to server

Problem is Recursive

Issue applies to all objects passed as:

- parameter

- return value

Applies to objects which are fields of objects listed above

Base Types

Usually base types are passed between client & server

What are base types in:

Java

Ruby

Python

Haskell

Smalltalk

Overloading Method Names

```
class RestaurantFinder {  
    Restaurant find(String name) { ... }  
    Restaurant find(int id) { ... }
```

Dynamic typed languages (Ruby, Python etc) don't support overloading

Rails & REST

The Hard Way

```
class BooksController < ApplicationController
```

```
  def booklist
```

```
    @book = Book.find(2)
```

```
  end
```

booklist.rhtml

```
<book>
```

```
  <title><%= @book.title %></title>
```

```
  <published_date><%= @book.date %></published_date>
```

```
  <publisher><%= @book.publisher.name %></publisher>
```

```
  <authors>
```

```
    <% for author in @book.authors %>
```

```
      <author>
```

```
        <first_name><%= author.first_name %></first_name>
```

```
        <last_name><%= author.last_name %></last_name>
```

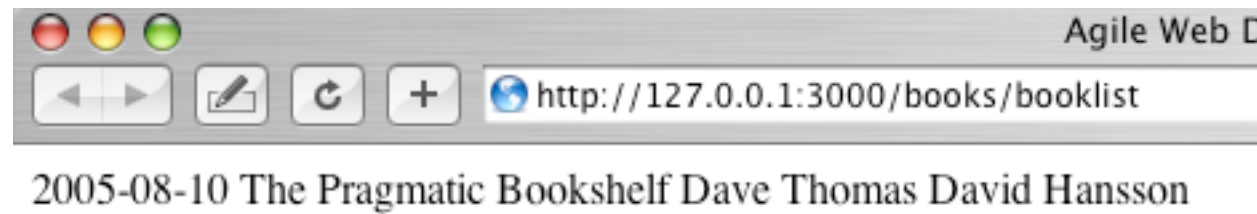
```
      </author>
```

```
    <% end %>
```

```
  </authors>
```

```
</book>
```

Response in Browser



Browsers differ in how they render XML

HTTP requests to server indicate types of response they accept

HTTP responses indicate the type of content

Response Headers

Cache-Control: no-cache

Connection: Keep-Alive

Date: Thu, 30 Nov 2006 18:32:07 GMT

Content-Type: text/html; charset=UTF-8

Server: WEBrick/1.3.1 (Ruby/1.8.2/2004-12-25)

Content-Length: 419

Set-Cookie: _session_id=241aa03ddc6464a03b7a982836485f15; path=/

200 OK

Response Source

```
<book>
  <title>Agile Web Development with Rails</title>
  <published_date>2005-08-10</published_date>
  <publisher>The Pragmatic Bookshelf</publisher>
  <authors>

    <author>
      <first_name>Dave</first_name>
      <last_name>Thomas</last_name>
    </author>

    <author>
      <first_name>David</first_name>
      <last_name>Hansson</last_name>
    </author>

  </authors>
</book>
```

Generating XML

```
class BooksController < ApplicationController
```

```
  def booklist
```

```
    @book = Book.find(2)
```

```
  end
```

booklist.rxml

```
xml.book do
```

```
  xml.title(@book.title)
```

```
  xml.published_date(@book.date)
```

```
  xml.publisher(@book.publisher.name)
```

```
  xml.authors do
```

```
    for author in @book.authors
```

```
      xml.author do
```

```
        xml.first_name(author.first_name)
```

```
        xml.last_name(author.last_name)
```

```
      end
```

```
    end
```

```
  end
```

```
end
```

booklist.rxml is called if

booklist.rhtml does not exist

Response in Browser (Firefox)

This XML file does not appear to have any style information associated with it.

```
- <book>
  <title>Agile Web Development with Rails</title>
  <published_date>2005-08-10</published_date>
  <publisher>The Pragmatic Bookshelf</publisher>
  - <authors>
    - <author>
      <first_name>Dave</first_name>
      <last_name>Thomas</last_name>
    </author>
    - <author>
      <first_name>David</first_name>
      <last_name>Hansson</last_name>
    </author>
  </authors>
</book>
```


Response Headers

Response Headers - <http://127.0.0.1:3000/books/booklist>

Cache-Control: no-cache

Connection: Keep-Alive

Date: Thu, 30 Nov 2006 19:32:48 GMT

Content-Type: application/xml

Server: WEBrick/1.3.1 (Ruby/1.8.2/2004-12-25)

Content-Length: 385

Set-Cookie: _session_id=c975d78a743a9de3c1689a71a4f7191e; path=

200 OK

Response Source

```
<book>
  <title>Agile Web Development with Rails</title>
  <published_date>2005-08-10</published_date>
  <publisher>The Pragmatic Bookshelf</publisher>
  <authors>
    <author>
      <first_name>Dave</first_name>
      <last_name>Thomas</last_name>
    </author>
    <author>
      <first_name>David</first_name>
      <last_name>Hansson</last_name>
    </author>
  </authors>
</book>
```

Returning HTML or XML

```
class BooksController < ApplicationController

  def booklist
    @book = Book.find(2)
    respond_to do |wants|
      wants.html      #renders booklist.rhtml if requested
      wants.xml       #renders booklist.rxml if requested
    end
  end
end
```

How to request HTML or XML

HTTP Accept Header

Client uses HTTP Accept header to indicate types it will accept

Accept: text/html, text/plain, application/xml

rohan -> curl -H "Accept: application/xml" http://127.0.0.1:3000/books/booklist

Cache & Examples

Rails does cache pages to improve performance

Adding/deleting booklist.rhtml does not clear the cache

So you may need to restart server to see the change

Yet Another Way

```
class BooksController < ApplicationController

  def booklist
    @book = Book.find(2)
    respond_to do |wants|
      wants.html
      wants.xml { render :xml => @book.to_xml(:include => :authors) }
    end
  end
end
```

Generate XML without booklist.rxml

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<book>
  <authors>
    <author>
      <id type="integer">1</id>
      <first-name>Dave</first-name>
      <author-id>1</author-id>
      <last-name>Thomas</last-name>
      <book-id>2</book-id>
    </author>
    <author>
      <id type="integer">2</id>
      <first-name>David</first-name>
      <author-id>2</author-id>
      <last-name>Hansson</last-name>
      <book-id>2</book-id>
    </author>
  </authors>
  <title>Agile Web Development with Rails</title>
  <date type="date">2005-08-10</date>
  <id type="integer">2</id>
  <publisher-id type="integer">1</publisher-id>
</book>
```

Implement to_xml in classes
to change default

Rails & SOAP/XML-RPC

Generating Web Service

```
AI 25->ruby script/generate web_service Bookstore find_all_books find_book_by_id  
create app/apis/  
exists app/controllers/  
exists test/functional/  
create app/apis/bookstore_api.rb  
create app/controllers/bookstore_controller.rb  
create test/functional/bookstore_api_test.rb
```

AI 26->

Generated Files

app/controllers/bookstore_controller.rb

```
class BookstoreController < ApplicationController
  wsd_service_name 'Bookstore'
```

```
  def find_all_books
  end
```

```
  def find_book_by_id
  end
end
```

app/apis/bookstore_api.rb

```
class BookstoreApi < ActionWebService::API::Base
  api_method :find_all_books
  api_method :find_book_by_id
end
```

Edited Files

app/controllers/bookstore_controller.rb

```
class BookstoreController < ApplicationController
  wsd_service_name 'Bookstore'
  web_service_scaffold :invoke

  def find_all_books
    Book.find(:all).map { |book| book.id}
  end

  def find_book_by_id(id)
    Book.find(id)
  end
end
```

app/apis/bookstore_api.rb

```
class BookstoreApi < ActionWebService::API::Base
  api_method :find_all_books,
    :returns => [[:int]]
  api_method :find_book_by_id,
    :expects => [{:book_id => :int}],
    :returns => [Book]
end
```

Controller Details

```
class BookstoreController < ApplicationController
  wsd_service_name 'Bookstore'
    # associates name with service

  web_service_scaffold :invoke
    #Generates web pages to try out XML-PRC & SOAP
    # :invoke name can be anything
    # name given used in url
```

Specifying the Methods

```
class BookstoreApi < ActionWebService::API::Base
  api_method :find_all_books,
    :returns => [[:int]]
  api_method :find_book_by_id,
    :expects => [{:book_id => :int}],
    :returns => [Book]
end
```

```
:returns => [returnType]
```

```
:expects => [{argument1Name => argument1Type},
             {argument2Name => argument2Type},
             etc ]
```

Parameter Specifiers

Base Types

`:int`
`:string`
`:base64` (for binary data)
`:bool`
`:time`
`:datetime`
`:date`

Classes

Subclass of
`ActiveRecord::Base`
`ActionWebService::Struct`

Arrays

`:expects => [{:foo => [:int]}],`

http://127.0.0.1:3000/bookstore/wsdl

URI for SOAP WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="Bookstore" xmlns:typens="urn:ActionWebService" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="urn:ActionWebService" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns="http://
schemas.xmlsoap.org/wsdl/">
  <types>
    <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:ActionWebService">
      <xsd:complexType name="Book">
        <xsd:all>
          <xsd:element name="id" type="xsd:int"/>
          <xsd:element name="title" type="xsd:string"/>
          <xsd:element name="date" type="xsd:dateTime"/>
          <xsd:element name="publisher_id" type="xsd:int"/>
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="IntegerArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
            <xsd:attribute wsdl:arrayType="xsd:int[]" ref="soapenc:arrayType"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:schema>
  </types>
```

http://127.0.0.1:3000/bookstore/invoke

API Methods for Bookstore

- [Book FindBookById\(int param0\)](#)
- [int\[\] FindAllBooks\(\)](#)

Find all

Method Invocation Details for *Bookstore#FindAllBooks*

Protocol:

SOAP

Invoke

[Back](#)

Method Invocation Result fo

Invocation took 0.039730 secc

Return Value:

```
[1, 3, 2]
```

Request XML:

http://127.0.0.1:3000/bookstore/invoke

Method Invocation Details for *Bookstore#FindBookById*

Protocol:

SOAP

Method Parameters:

Param0 (int)

1

Invoke

[Back](#)

hour is a private method in Date
results in error

NoMethodError in BookstoreC

```
private method `hour' called for #<Date: 4906743/2,0,2299161>
```

XML-RPC Works

Method Invocation Details for *Bookstore#FindBookById*

Protocol:

XML-RPC

Method Parameters:

Param0 (int)

1

Invoke

[Back](#)

Returned XML

```
<?xml version="1.0" ?> <methodresponse> <params> <param> <value> <struct> <member>
<name> title </name> <value> <string> Programming Ruby </string> </value> </member>
<member> <name> date </name> <value> <datetime.iso8601> 20050101T00:00:00 </
datetime.iso8601> </value> </member> <member> <name> id </name> <value> <i4> 1 </i4> </
value> </member> <member> <name> publisher_id </name> <value> <i4> 1 </i4> </value> </
member> </struct> </value> </param> </params> </methodresponse>
```

Fixing the Soap Version

app/models/bookstruct.rb

```
class Bookstruct < ActionWebService::Struct
  member :title, :string
  member :publisher, :string
  member :authors, [:string]
  member :date, :string

  def from_book(book)
    @title = book.title
    @publisher = book.publisher.name
    @authors = []
    for author in book.authors do
      @authors << (author.last_name)
    end
    @date = book.date.to_s
  end
end
```

Fixing the Soap Version

app/controllers/bookstore_controller.rb

```
class BookstoreController < ApplicationController
  wsdl_service_name 'Bookstore'
  web_service_scaffold :invoke

  def find_all_books
    Book.find(:all).map { |book| book.id}
  end

  def find_book_by_id(id)
    book = Bookstruct.new
    book.from_book(Book.find(id))
    book
  end
end
```

Fixing the Soap Version

app/apis/bookstore_api.rb

```
class BookstoreApi < ActionWebService::API::Base
  api_method :find_all_books,
    :returns => [[:int]]
  api_method :find_book_by_id,
    :expects => [{:book_id => :int}],
    :returns => [Bookstruct]
end
```

Using XML-RPC

```
require 'xmlrpc/client'  
client = XMLRPC::Client.new("127.0.0.1", "/bookstore/api", 3000)  
puts client.call("FindAllBooks")  
puts client.call("FindBookById", 2)
```

Ruby SOAP Client did not work

```
require 'soap/wsdlDriver'  
require 'cgi'
```

```
WSDL_URL = 'http://127.0.0.1:3000/bookstore/wsdl'
```

```
client = SOAP::WSDLDriverFactory.new(WSDL_URL).createDriver  
result = client.FindAllBooks
```

Error Message

```
NoMethodError: undefined method 'FindAllBooks' for #<SOAP::WSDLDriver:  
{urn:ActionWebService}BookstoreBookstorePort>
```

Smalltalk SOAP Client Works

client := BookstoreBookstorePortClient1 new.

book := client findBookById: 2.

allBooks := client findAllBooks.

Rails Functional SOAP Tests did not work

test/functional/bookstore_api_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
require 'bookstore_controller'
```

```
class BookstoreController; def rescue_action(e) raise e end; end
```

```
class BookstoreControllerApiTest < Test::Unit::TestCase
  def setup
    @controller = BookstoreController.new
    @request     = ActionController::TestRequest.new
    @response    = ActionController::TestResponse.new
  end
```

```
  def test_find_all_books
    result = invoke :find_all_books
    assert result[0].is_a?(Integer)
  end
```

```
  def find_book_by_id
    result = invoke :find_book_by_id, 2
    assert_equal nil, result
  end
end
```