

# Rails and Web Services

The Client Side

The Easier Way

**WS-**





Consuming

SOAP / WSDL

XML-RPC

REST

SOAP / WSDL complex

XML-RPC less complex

REST simple

# SOAP / WSDL

SOAP = Simple Object Access Protocol

WSDL = Web Services Description Language

```
Microsoft Development Environment [design] - GoogleSearch.wsdl
File Edit View Debug Tools Window Help
SMTPServer
GoogleSearch.wsdl
<?xml version="1.0"?>
<!-- WSDL description of the Google Web APIs.
The Google Web APIs are in beta release. All interfaces are subject to
change as we refine and extend our APIs. Please see the terms of use
for more information. -->
<!-- Revision 2002-08-16 -->
<definitions name="GoogleSearch"
targetNamespace="urn:GoogleSearch"
xmlns:types="urn:GoogleSearch"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<!-- Types for search - result elements, directory categories -->
<types>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:GoogleSearch">
<xsd:complexType name="GoogleSearchResult">
<xsd:all>
<xsd:element name="documentFiltering" type="xsd:boolean"/>
<xsd:element name="searchComments" type="xsd:string"/>
<xsd:element name="estimatedTotalResultsCount" type="xsd:int"/>
<xsd:element name="estimateIsExact" type="xsd:boolean"/>
<xsd:element name="resultElements" type="types:ResultElementArray"/>
<xsd:element name="searchQuery" type="xsd:string"/>
<xsd:element name="startIndex" type="xsd:int"/>
<xsd:element name="endIndex" type="xsd:int"/>
<xsd:element name="searchTips" type="xsd:string"/>
<xsd:element name="directoryCategories" type="types:DirectoryCategoryArray"/>
<xsd:element name="searchTime" type="xsd:double"/>
</xsd:all>
</xsd:complexType>
<xsd:complexType name="ResultElement">
<xsd:all>
<xsd:element name="summary" type="xsd:string"/>
<xsd:element name="URL" type="xsd:string"/>
<xsd:element name="snippet" type="xsd:string"/>
<xsd:element name="title" type="xsd:string"/>

```

Ready Ln 50 Col 69 Ch 69 INS

Ln 20 Col 68 Ch 68 INS

<xsd:element name="documentFiltering" type="xsd:boolean"/>

# SOAP / WSDL Example

Soap4r

XSD4R

require 'soap/wsdlDriver'

# Keep it simple

.NET generated WSDL != Ruby Soap4r

do I need to send a 'simple' object down the wire?

my experience

# XML-RPC

eXtensible Markup Language - Remote Procedure Call

## XML-RPC Request Format

XML-RPC requests are specially formatted XML data posted to a URL. You can find out more about XML-RPC at [www.xmlrpc.com/](http://www.xmlrpc.com/).

The XML-RPC Server Endpoint URL is <http://www.flickr.com/services/xmlrpc/>

To request the flickr.test.echo service, send a request like this:

```
<methodCall>
  <methodName>flickr.test.echo</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>name</name>
            <value><string>value</string></value>
          </member>
          <member>
            <name>name2</name>
            <value><string>value2</string></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Note that the method name is sent separately from the other request parameters.

If a parameter is specified, it should be a single struct member with the above formatting.

By default, XML-RPC requests will send a [XML-RPC response](#).

By default, XML-RPC requests will send a [XML-RPC response](#).

If a parameter is specified, it should be a single struct member with the above formatting.

Note that the method name is sent separately from the other request parameters.

# XML-RPC Example

require 'xmlrpc/client'

# REST

REpresentational State Transfer

find

create

update

destroy

find

create

update

destroy

SELECT

INSERT

UPDATE

DELETE

GET	POST	PUT	DELETE
find	create	update	destroy
SELECT	INSERT	UPDATE	DELETE

# Custom XML

# REST Example

The Easier Way

# Providing

1.1.4

# Roll your own REST

.rxml file and corresponding method  
disable the layout for REST methods  
'gem install builder'

# ActionWebService

`'script/generate web_service service_name method_name'`

support for both SOAP and XML-RPC

dynamic WSDL

Planning

`'script/plugin install actionwebservice'`

REST

CRUD

ActiveResource

# CRUD

```
class PeopleController < ActionController::Base
  # GET /people
  def index() end

  # GET /people/new
  def new() end

  # POST /people
  def create() end

  # GET /people/1
  def show() end

  # GET /people/1/edit
  def edit() end

  # PUT /people/1
  def update() end

  # DELETE /people/1
  def destroy() end
end
```

# ActiveResource

Person.find(1).name => 'Phil'

GET <http://my.api.com/people/1>

```
<person><name>Phil</name></person>
```



iconbuffet

Welcome to IconBuffet. Your cart has 0 items

Login

Icons | Illustration | Free Delivery

# MANHATTAN



Pretty much our most popular icons, ever. Gosh!

Join thousands, and **get free icons** from IconBuffet every month!

The icons look really good, but better than that, they make us look really good too.

—Jim Coudal, Coudal.com

**Oslo Buzz**  
on sale this month for five bucks.



Includes PNG, GIF and EPS Vector files!

#### Browse Our Products

- [Standard Icons](#)
- [Blog & CMS Icons](#)
- [Starter Icons](#)
- [IconBuffet Vectors](#)
- [Holiday Vectors](#)
- [Holiday Icons](#)

## Get Free Icons Every Month with IconBuffet Free Delivery.

Each month IconBuffet Free Delivery members receive free stock icons. There's absolutely no cost to you, and the icons are royalty-free for use in commercial and personal products. [Sign](#)

#### New Products



Kyoto Vector

Kyoto Vector



# Sources

DHH RailsConf Keynote - <http://tinyurl.com/opl7c>

Kevin Marshall - Web Services on Rails PDF

Ryan Daigle - <http://tinyurl.com/oq87k>

Scott Raymond - <http://ScottRaymond.net>



[pdatasolutions.com](http://pdatasolutions.com)