Web services
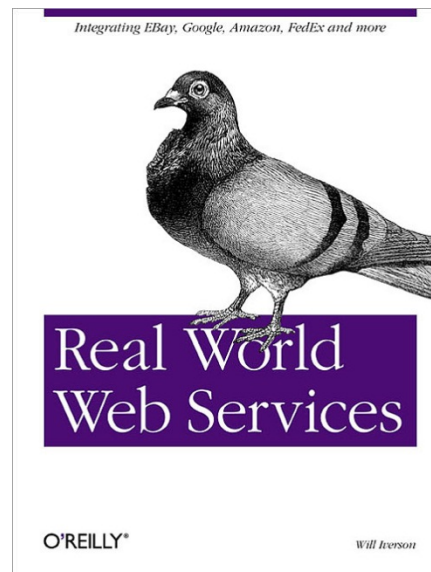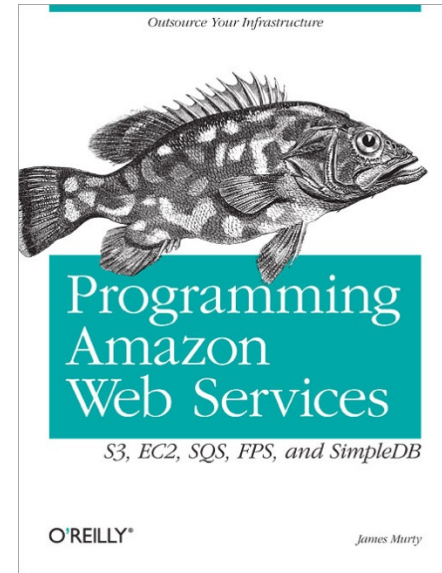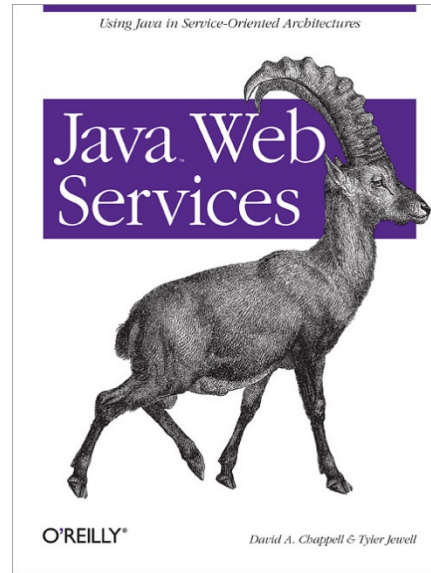
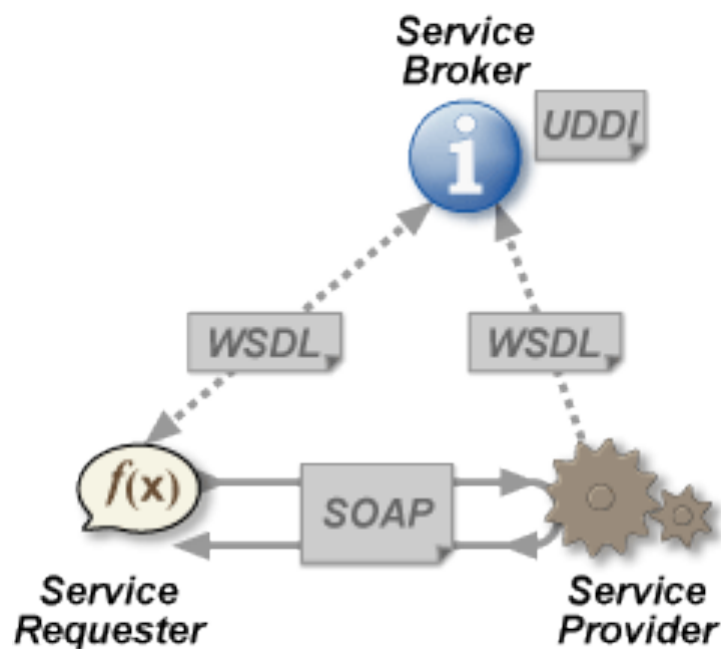# Overview

- Web services
  - What does that mean?
  - Why are they useful?
    - Examples!
- Major interaction types
  - REST
  - SOAP
- Data formats:
  - XML
  - JSON

Web service

From Wikipedia, the free encyclopedia

A **Web service** is a method of communication between two electronic devices over the web.

W3C says...

1.4 What is a Web service?

For the purpose of this Working Group and this architecture, and without prejudice toward other definitions, we will use the following definition:

[Definition: A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.]

Web services

- Basic idea:
  - Allows others to make use of your:
    - Unique functionality, e.g. translating English to Spanish
    - Unique data, e.g. find out where your FedEx package is
  - Do this over the Internet
    - In a standard way using a known protocol (e.g. HTTP)
  - Possible business uses:
    - Within a company to integrate things
    - Between a company and partners
    - For free, promote your new fangled search engine (e.g. Bing)
    - For money, (e.g. $5/1000 search queries)

Bing web services

Develop with Bing

Use Bing Search and Map APIs to increase your website's functionality and user appeal, or to create enterprise/consumer applications.

Sign in - Bing Search API	Sign in - Bing Maps
Sign up to use Bing Search API and create ApplDs Learn more	Sign up to use Bing Maps Learn more

Create unique search applications

- Build applications powered by Bing's technology
- Choose from multiple SourceTypes (Web, Images, Video, and More) and output protocols (JSON, SOAP, or XML)
- Customize the search results to your needs

Create your own mapping applications

- Feature maps that render fast and deliver stunning imagery
- Easily scale from a simple static map up to the most complex interactive spatial application
- Build for mobile using our tools for Windows Phone 7, Android, and iOS

Using Bing search API

- Apply for an app ID
  - Get a Windows Live account
  - e.g. ABDJ3983HEFHB39398FEFE37
- Choose your "protocol":
  - JavaScript Object Notation (JSON)
  - Extensible Markup Language (XML)
  - SOAP (original Simple Object Access Protocol)
- Make your search request
  - Using a language/command line tool of your choice
  - My example: REST with JSON result format

Using Bing search API

- Find the top-10 Bing results for "orediggers"
  - Make a HTTP GET request
  - `http://api.bing.net/json.aspx?`
    ```
    AppId=AFKJEAWKFJEAWKFJA&Version=2.2&
    Market=enUS&
    Query=orediggers&Sources=web+spell&
    Web.Count=10&
    JsonType=raw
    ```

Yahoo web services

BOSS API

Yahoo! Search BOSS API

BOSS (Build your Own Search Service) is Yahoo!'s open search and data services platform. The goal of BOSS is simple: to foster innovation in the search industry. Developers, start-ups, and large Internet companies can use BOSS to build web-scale search products that utilize Yahoo! Search technology and data. By combining your unique assets and ideas with our search technology assets, you can build innovative experiences that delight your users. BOSS is offered with a low flexible usage fee based on the type of queries. You can also get search advertising in the same service to monetize your offerings.

BOSS + Data / Ideas / Content → Search

Your Special Sauce   **Innovative Search Experience**

Google web services

Google Latitude API

What is the Google Latitude API?

The Google Latitude API allows for websites and programs to integrate with Google Latitude, enabling users to update and read their current location, their location history, and more!

How do I start?

1. Find out the basics on Getting Started.
2. Find out how to use REST to invoke the Google Latitude API by reading the Developer's Guide.
3. Browse the Google Latitude API Reference Guide.
4. Get community support. Join our community and participate in our discussion group.

Google Calendar APIs and Tools

What are the Google Calendar APIs and Tools?

Google calendar

Life's important events all in one place. Google Calendar offers many ways to create and share content other than the web interface that we all know and love.

Calendar API

The Calendar API lets you incorporate Calendar functionality into your own application or website. You can edit calendars, create and delete events, send invitations, and more.

Calendar Gadgets

Calendar Gadgets let you extend Google Calendar to give your users a custom, content-rich experience. Create status displays, interactive events, and custom user interface controls.

☆ **JSON/Atom Custom Search API**

Developer's Guide

Programmer's Guides

JSON/Atom Custom Search API

The JSON/Atom Custom Search API lets you develop websites and programs to retrieve and display search results from Google Custom Search programmatically. With this API, you can use RESTful requests to get search results in either JSON or Atom format.

Important: The JSON/Atom Custom Search API requires the use of an API key, which you can get from the Google APIs console. The API provides 100 search queries per day for free. If you need more, you may sign up for billing in the console.

Available Documentation

Version 1 is the latest available version of the JSON/Atom Custom Search API. Its documentation includes:

- A **Developer's Guide**. Provides information on basic concepts and step-by-step instructions on how to use the common features of the API.

  This guide focuses on the RESTful method of invoking the API, or HTTP calls that use REST verbs (such as GET and POST) to access JSON or Atom data structures.

You can access the documentation available for the JSON/Atom Custom Search API below:

API Version	Developer's Guide	Reference
v1	Getting Started, Using REST	Reference

Pricing

Free quota

Usage is free for all users, up to 100 queries per day.

Paid Usage

Any usage beyond the free usage quota will fail if you are not signed up for billing. Once you have enabled billing, you will continue to receive 100 free queries per day. However, you will be billed for all additional requests at the rate of $5 per 1000 queries, for up to 10,000 queries per day. If you need additional quota, please request additional quota from the console.

Facebook web services

FedEx web services

Twitter web services

twitter developers   Search   API Health   Blog   Discussions   Docum

Home → Documentation

Getting Started

Updated on Wed, 2011-09-28 16:40

Twitter is an information network and communication mechanism that produces more than 200 million tweets a day. The Twitter platform offers access to that corpus of data, via our APIs. Each API represents a facet of Twitter, and allows developers to build upon and extend their applications in new and creative ways. It's important to note that the Twitter APIs are constantly evolving, and developing on the Twitter Platform is not a one-off event.

Sipping on the Twitter Spritzer…

`statuses/sample`

Returns a random sample of all public statuses. The default access level, 'Spritzer' provides a small proportion of the Firehose, very roughly, 1% of all public statuses. The "Gardenhose" access level provides a proportion more suitable for data mining and research applications that desire a larger proportion to be statistically significant sample. Currently Gardenhose returns, very roughly, 10% of all public statuses. Note that these proportions are subject to unannounced adjustment as traffic volume varies.

- **URL:** *https://stream.twitter.com/1/statuses/sample.json*
- **Method(s):** GET
- **Parameters:** count, delimited, stall_warnings
- **Returns:** stream of status_element

Just go to this URL in a browser and enter your Twitter username and password! Or programmatically:
```
curl -k https://stream.twitter.com/1/statuses/sample.json -umyuser:mypassword
```

Programmatic access...

```csharp
static void Main(string[] args)
{
    HttpWebRequest   webRequest      = null;
    HttpWebResponse webResponse      = null;
    StreamReader     responseStream  = null;

    while (true)
    {
        try
        {
            webRequest = (HttpWebRequest)
              WebRequest.Create("https://stream.twitter.com/1/statuses/sample.json");
            webRequest.Credentials = new NetworkCredential("username", "password");
            webRequest.Timeout = -1;
            webResponse = (HttpWebResponse) webRequest.GetResponse();
            responseStream = new StreamReader(webResponse.GetResponseStream(),
                               System.Text.Encoding.GetEncoding("utf-8"));
            Console.WriteLine(responseStream.ReadLine());
        }
        catch (WebException ex)
        {
            Console.WriteLine(ex.Message);
        }
...
```

C# example printing out the Twitter spritzer.

Twitter streaming API

- Streaming API: ~1% of the world's Tweets for free
  - Since early 2011:
    - 115 GB compressed (ibex)

```
{"retweeted":false,"text":"@cellulose_gum quase sempre","geo":null,"created_at":"Mon Dec 20 03:32:4
{"retweeted":false,"text":"@Ferestart123 porque eu n\u00e3o falo com meu namorado desde quarta :S",
{"retweeted":false,"text":"RT @operationSAFE: YW @butterfly101664 Thanks for the #opsafe love! #tea
{"retweeted_status":{"retweeted":false,"text":"Vote for @Rihanna and @RihannaDaily http:\/\/www.z10
{"retweeted":false,"text":"RT @iTz_MaMa_2U: Happy Birthday Sophie !! @PurpleAlamode Enjoy your day
{"retweeted":false,"text":"I just bought a 180sx and just wondering what problems you guys have had
{"retweeted":false,"text":"concerveja n\u00e3o \\/ (@withoutnotion live on http:\/\/twitcam.com\/33
{"retweeted":false,"text":"@MoreBradley Maybe we should start #needsmorebradleystache","geo":null,"
{"retweeted":false,"text":"q merda, n consigo ir pro banho.. o papo ta mt interessante @_@","geo":n
{"retweeted":false,"text":"Ain't no body as cold as this do the rap and the track. Triple Double No
{"retweeted":false,"text":"My bad. ████ shut up.","geo":null,"created_at":"Mon Dec 20 03:32:45 +000
{"retweeted":false,"text":"@u_kisseli what is \uc57c\uc6e5\uc774? U gave him away? Don't be sad \uc
{"retweeted":false,"text":"Acabo de completar \"Qual o seu tipo de sorriso?\" e meu resultado foi:
{"retweeted":false,"text":"Tuh @adiTIAN_ , jgn suka nambah sambel ya!! RT @detikcom: Harga Cabai Ki
{"retweeted_status":{"retweeted":false,"text":"Como no encuentras \"con qui\u00e9n\", te obsesionas
{"retweeted_status":{"retweeted":false,"text":"There is no generational divide on Social Security:
{"retweeted":false,"text":"Ahora a apagar la TV hasta que vuelva la 4ta temporada de #los80.","geo"
{"entities":{"user_mentions":[{"indices":[5,21],"id_str":"18697486","name":"Flip","id":18697486,"sc
{"retweeted":false,"text":"Developing in a 'Final Frontier' of Hoboken: EVEN in a tough real estate
```

I just bought some milk…

- **What do with all this Twitter data?**
  - Tweets tend to be informal person-to-person like communications
  - Augmentative and Alternative Communication (AAC)
    - Enable users with certain disabilities to participate in everyday conversations
    - AAC devices often rely on statistical language models
    - Language models historically have been trained poorly on non-representative data

The Imagination of Crowds: Conversational AAC Language Modeling using Crowdsourcing and Large Data Sources
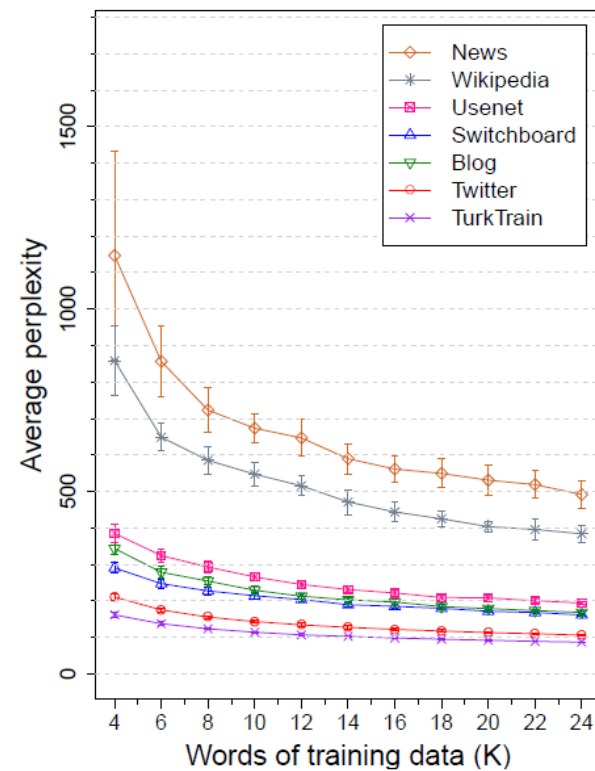
Abstract

Augmented and alternative communication (AAC) devices enable users with certain communication disabilities to participate in everyday conversations. Such devices often rely on statistical language models to improve text entry by offering word predictions. These predictions can be improved if the language model is trained on data that closely reflects the style of the users' intended communications. Unfortunately, there is no large dataset consisting of genuine AAC messages. In this paper we demonstrate how we can crowdsource the creation of a large set of fictional AAC messages. We show that these messages model conversational AAC better than the currently used datasets based on telephone conversations or newswire text. We leverage our crowdsourced messages to intelligently select sentences from much larger sets of Twitter, blog and Usenet data. Compared to a model trained only on telephone transcripts, our best performing model reduced perplexity on three test sets of AAC-like communications by 60–82% relative. This translated to a potential keystroke savings in a predictive keyboard interface of 5–11%.
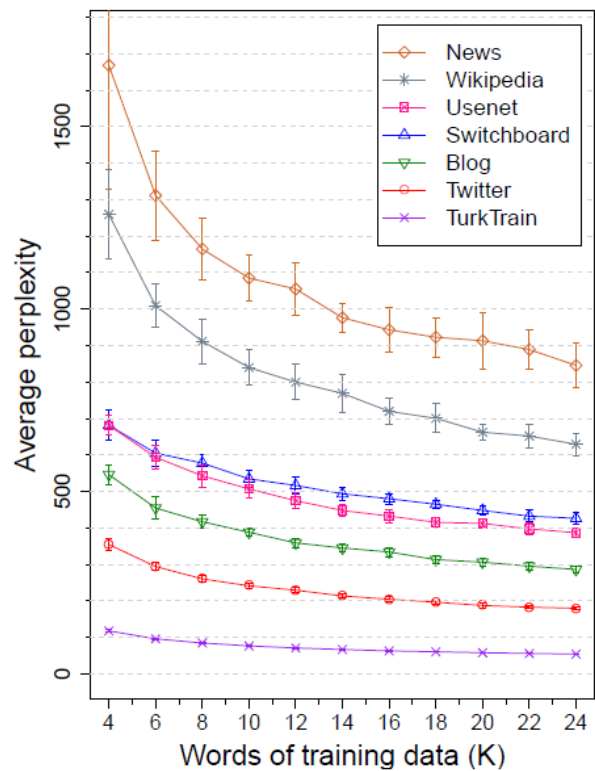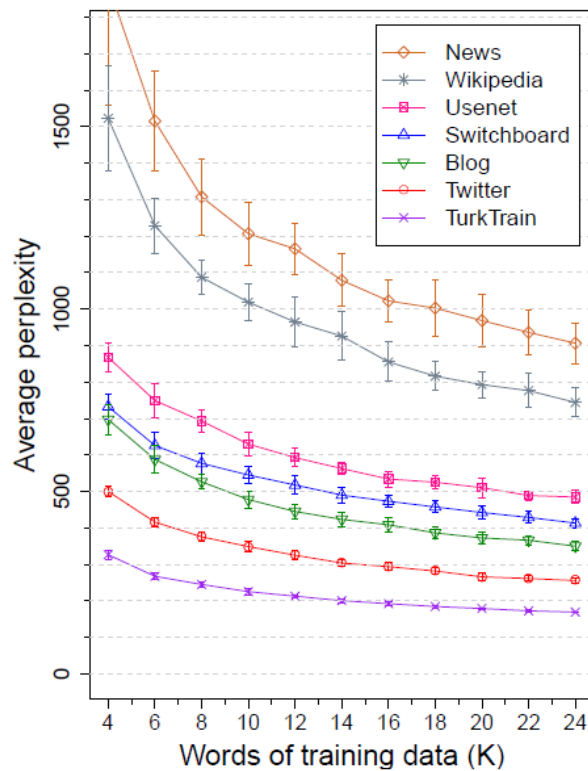
(a) TURKDEV test set (b) COMM test set (c) SPECIALISTS test set
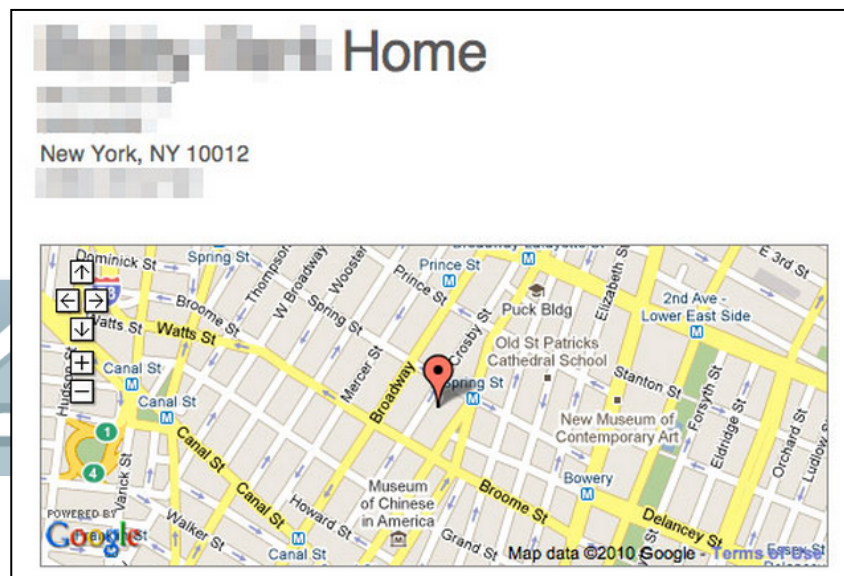
Mashups

- ## Mashups
  - A web application hybrid
    - Combine the functionality or data from several web sites
  - Frequently done using web services
    - e.g. Combine Google Maps API with Twitter API to show

PLEASE ROB ME

Raising awareness about over-sharing

Check out our guest blog post on the CDT website.

Home

New York, NY 10012

Web services protocols

- **Two major protocols:**
  - REST (Representational state transfer)
    - An HTTP GET request to a specific URL
    - HTTP is the protocol, no other choice
    - e.g. my Bing and Twitter examples
  - SOAP
    - Originally Simple Object Access Protocol
      - Dropped acronym, not so simple?
    - XML message format
    - Really a framework for specifying protocols
      - HTTP is one profile choice
    - Strong typing
      - Generate proxy class using toolkit

Bing via SOAP

Using SOAP (Bing, Version 2)

msdn

6 out of 7 rated this helpful   Rate this topic

Bing Services

The Bing SOAP interface is most efficiently accessed by referencing the ***Web Service Description Language*** (WSDL) document from a Microsoft Visual Studio project. The WSDL defines the ports and messages that comprise the Bing API SOAP web service.

To add a Web reference in Microsoft Visual Studio

1. From **Solution Explorer** in an existing or newly created project, right-click **References** and, from the pop-up menu, select **Add Service Reference**.

   If you are using Microsoft Visual Studio 2005, this pop-up menu includes **Add Web Reference**. In this case, click **Add Web Reference** and proceed to Step 3.

   If you are using Microsoft Visual Studio 2008, proceed to Step 2.

2. Click **Advanced** on the **Add Service Reference** dialog box, then click **Add Web Reference** on the **Service Reference Settings** dialog box.

3. Type the following address in the URL text box: **http://api.bing.net/search.wsdl?AppID**=*YourAppId***&Version=2.2**. For information about obtaining an AppId, see Bing Developer Center.

4. Click **Go.**

5. You can accept the default web reference name `net.bing.api` suggested in the **Web reference name** text box, or type your own name for the web reference in the text box. Click **Add Reference** to add the web reference to your project.

23

```csharp
using BingSOAP.net.bing.api;

namespace BingSOAP
{
    class Program
    {
        static void Main(string[] args)
        {

            BingService service = new BingService();
            SearchRequest request = new SearchRequest();

            request.AppId = "FAEWKJAEAEFJKAFWJKJAEFKJEFWKAFEWJKAWEFKAFWEJFAWE";
            request.Sources = new SourceType[] { SourceType.Web };
            request.Query = "orediggers";

            SearchResponse response = service.Search(request);
            int i = 0;
            foreach (WebResult r in response.Web.Results)
            {
                Console.WriteLine(i + ": " + r.Title);
                Console.WriteLine(i + ": " + r.Url);
                Console.WriteLine(i + ": " + r.Description);
                Console.WriteLine();
                i++;
            }
        }
    }
}
```

C# example that does a query using the Bing SOAP API.

```
Administrator: cmd                                              [_] [□] [✕]

c:\Dropbox\mtech\websci\BingSOAP\bin\Debug>BingSOAP.exe
0: Colorado School of Mines Athletics
0: http://www.csmorediggers.com/
0: Official site of the Orediggers with scores, statistics, pictures, and roster
s.

1: Colorado School of Mines Athletics
1: http://www.csmorediggers.com/landing/index
1: No. 18 Mountaineers Too Much For Orediggers, 28-6 January 28, 2012 Eighth-ran
ked wrestlers Steven Kelly and Jordan Larsen each posted wins, but visiting and
18th ...

2: Montana Tech Athletics
2: http://www.godiggers.com/
2: Official site of the Orediggers with news items, scores, statistics, player p
rofiles, roster and a schedule of games.

3: Colorado School of Mines - Wikipedia, the free encyclopedia
3: http://en.wikipedia.org/wiki/Colorado_Mines_Orediggers_football
3: The Colorado School of Mines (CSM, also referred to as "Mines") is a small pu
blic teaching and research university devoted to engineering and applied science
, with ...
```

"REST vs SOAP"

About 79,900 results (0.29 seconds)

Roots of the REST/SOAP Debate
www.prescod.net/rest/rest_vs_soap_overview/
Abstract. In order to communicate over networks we need standardized data formats and protocols. But how do we move forward toward this goal? One popular ...

"SOAP sucks"

About 32,100 results (0.10 seconds)

Why SOAP sucks - somebits.com
www.somebits.com/weblog/tech/bad/whySoapSucks.html
Nov 17, 2006 – Why SOAP sucks. There's an amusing dialogue floating around about how simple SOAP is. As someone who bears some past responsibility for ...

SOAP vs. REST | Steve Francia's Epic Blog
spf13.com/post/soap-vs-rest
Jan 15, 2010 – A good read is Why SOAP Sucks by Nelson Minar linked above. He was the guy that implemented Google's SOAP interface, which was often ...
You've visited this page 2 times. Last visit: 1/31/12

Your Soap Sucks - YouTube
www.youtube.com/watch?v=XIKFORGW3Ig
Oct 22, 2009 - 4 min - Uploaded by communitychannel
can bite me. Let me know what the best/most practical gift you ▶ 4:14 have ever received is! Hope you're all well and ...

More videos for **"SOAP sucks"** »

Data format: XML

- **XML (Extensible Markup Language)**
  - Human readable
  - W3C recommendation
  - Markup language like HTML
  - Originally a document format
    - Microsoft Office (Office Open XML)
    - OpenOffice (OpenDocument)
    - Apple iWork
  - Can be used for data interchange
    - e.g. SOAP, RSS, Atom

```
<?xml version="1.0"?>
<quiz>
 <question>
 Who was the forty-second
 president of the U.S.A.?
 </question>
<answer>
 William Jefferson Clinton
</answer>
<!-- Note: We need to add
 more questions later.-->
</quiz>
```

XML

XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

root element (required)

child elements

Every tag must be closed

```
<b><i>This text is bold and italic</b></i>
```

Example of improperly nested HTML

```
<b><i>This text is bold and italic</i></b>
```

Properly nested XHTML

```xml
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
  <!-- This is a comment -->
</bookstore>
```
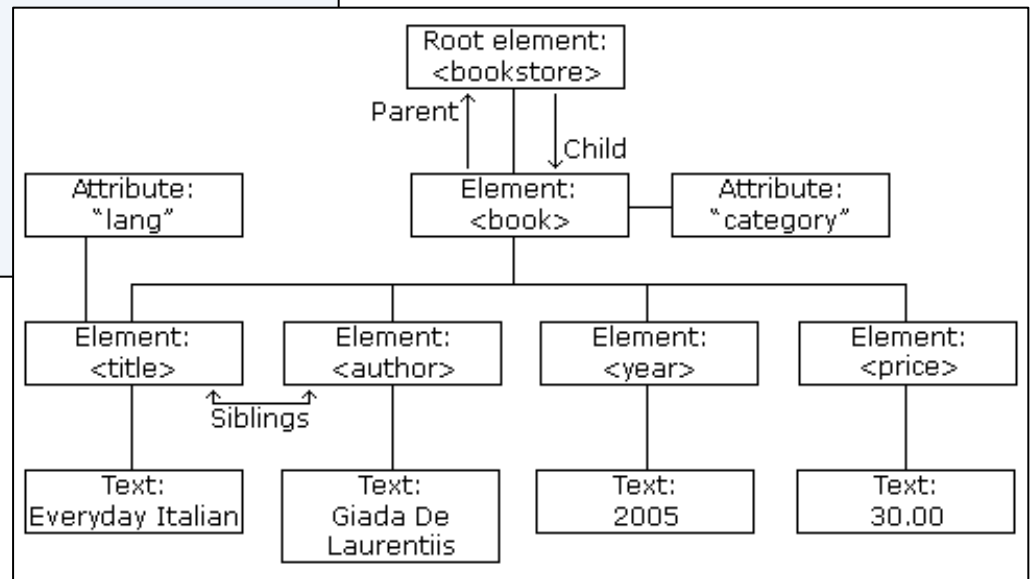
Tags are case-sensitive, <title> != <Title>

Attributes must be quoted

A comment in the XML document

```
<p>This is a paragraph.
<br>
```

Unclosed tags in an HTML document.

```
<p>This is a paragraph.</p>
<br />
```

In XHTML, every tag must be closed.

Whitespace is preserved

```
<message>if     salary &lt; 1000 then</message>
```

Special characters need to be escaped.

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

```
<note date="10/01/2008">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```
<note>
  <date>10/01/2008</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```
<note>
  <date>
    <day>10</day>
    <month>01</month>
    <year>2008</year>
  </date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Validating XML

- **Method 1: Document Type Definition (DTD)**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

Validating XML

- Method 2: XML Schema
  - XSD (XML Schema Defintion)

```
<xs:element name="note">

<xs:complexType>
  <xs:sequence>
    <xs:element name="to"      type="xs:string"/>
    <xs:element name="from"    type="xs:string"/>
    <xs:element name="heading" type="xs:string"/>
    <xs:element name="body"    type="xs:string"/>
  </xs:sequence>
</xs:complexType>

</xs:element>
```

Parsing XML

- **XML DOM (Document Object Model)**
  - Expressing and interacting with XML data
  - Modern browsers/languages have a built-in DOM parser
    - libxml2
      - Bindings for C++, C#, Python, PHP, Perl, …
    - MSXML
      - Microsoft XML Core Services

- **Or, just use simple string parsing**
  - Find stuff between start and end tag
  - Probably naughty… but fast…

```
<html><body><h1>W3Schools Internal Note</h1>
<div>
<b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span>
</div>

<script type="text/javascript">
if (window.XMLHttpRequest)
{ // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp = new XMLHttpRequest();
}
else
{ // code for IE6, IE5
  xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.open("GET","note.xml",false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;

document.getElementById("to").innerHTML =
    xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML =
    xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML =
    xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>
</body></html>
```
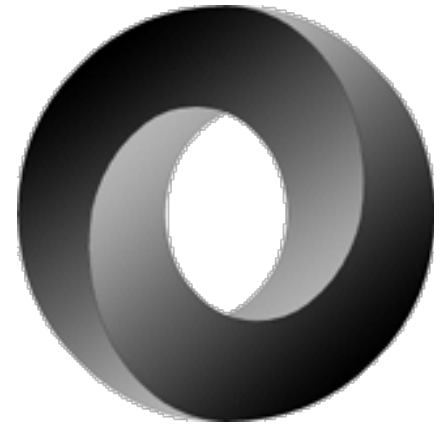
Data format: JSON

- **JSON (JavaScript Object Notation), "Jason"**
  - 2001 Popularized by Doug Crockford, Yahoo
    - http://youtu.be/-C-JoyNuQJs
  - Lightweight alternative to XML
    - Spec fits on the back of a business card
  - Human readable
  - Derived from how JavaScript represents data structures and associative arrays
 - Language independent data interchange

```
{to:"session", do:"test", text:"Hello world"}
```
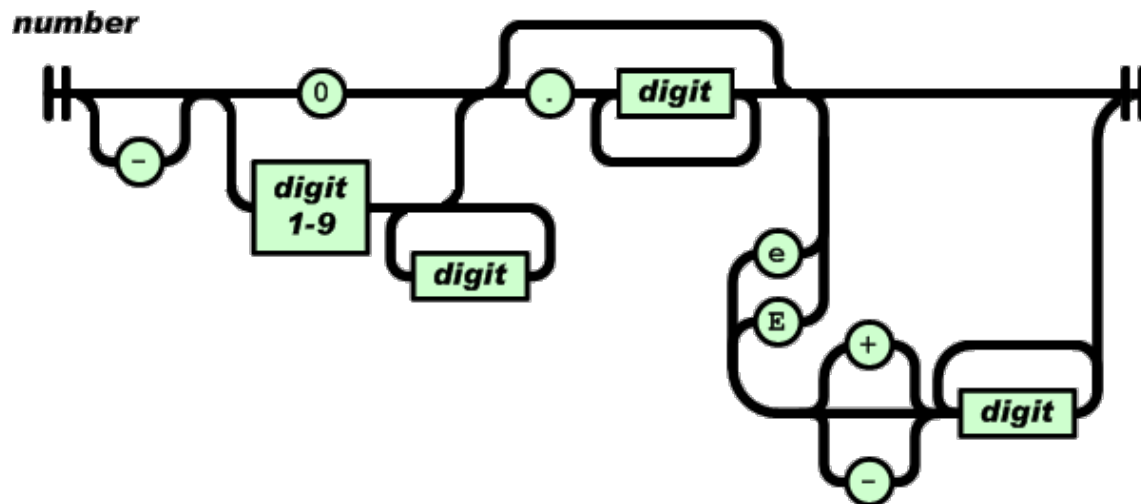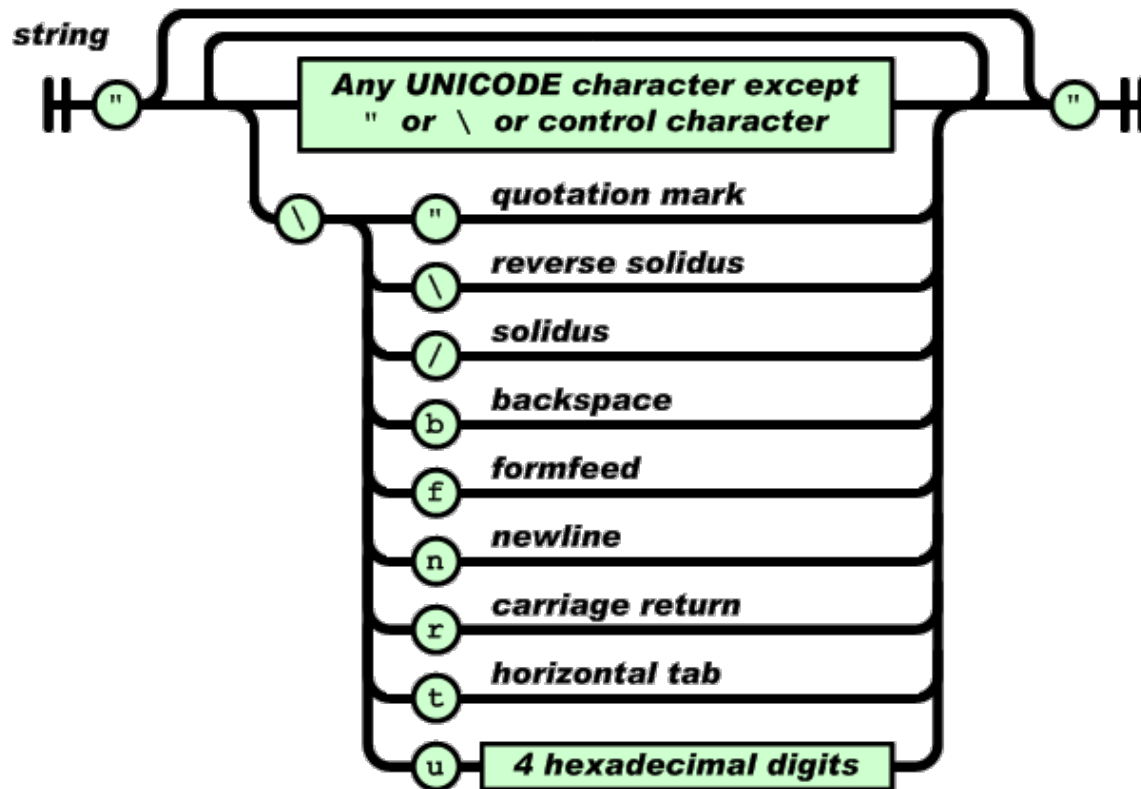
The world's first JSON message.
Failed due to JavaScript reserved word "do".
Decided to always force quoting of key.

JSON example

```json
{
    "firstName": "John",
    "lastName" : "Smith",
    "age"      : 25,
    "address"  :
    {
        "streetAddress": "21 2nd Street",
        "city"         : "New York",
        "state"        : "NY",
        "postalCode"   : "10021"
    },
    "phoneNumber":
    [
        {
          "type"  : "home",
          "number": "212 555-1234"
        },
        {
          "type"  : "fax",
          "number": "646 555-4567"
        }
    ]
}
```

JSON's grammar

string

number
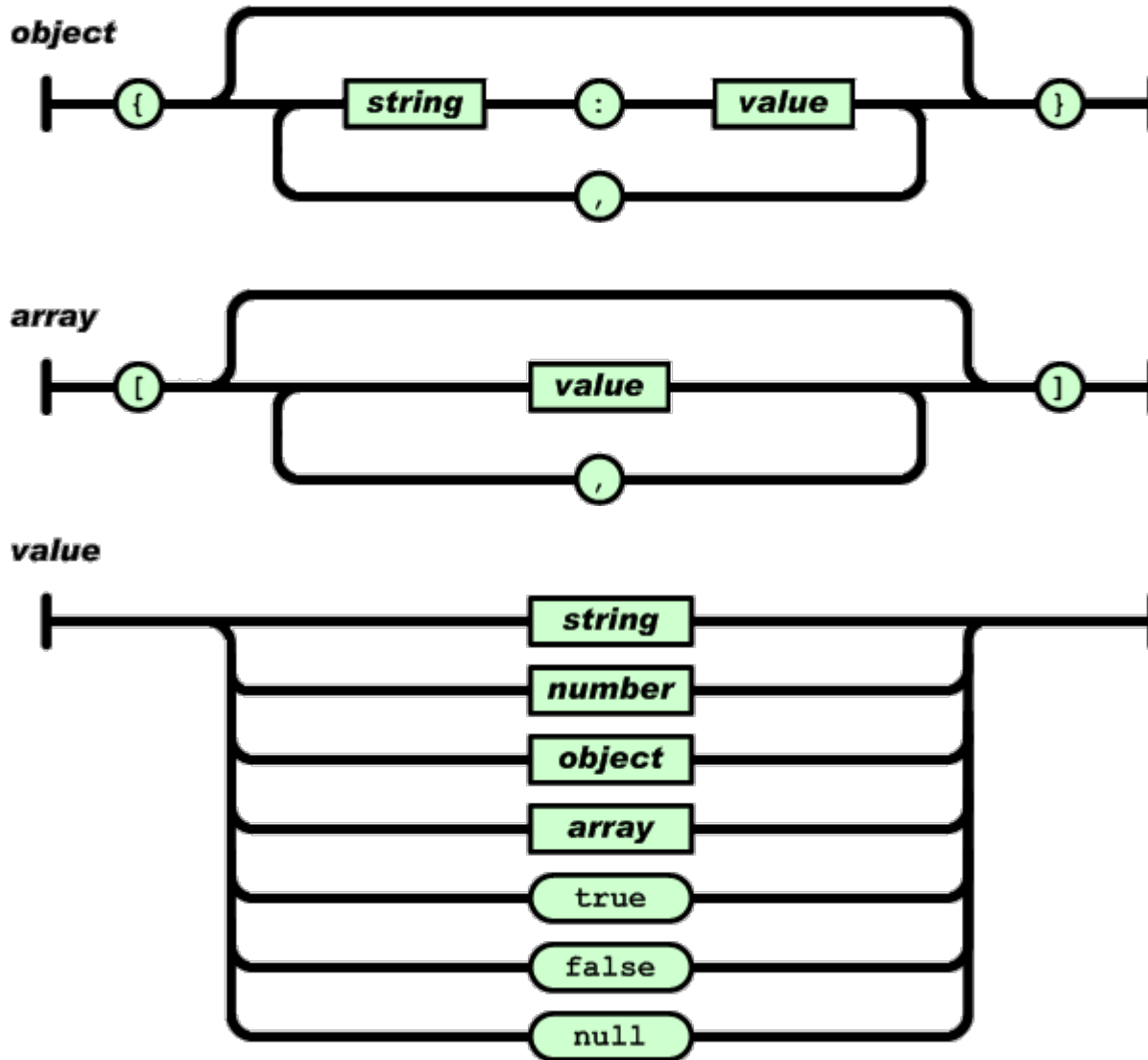
Parsing JSON

- In JavaScript
  - JSON is subset of the literal JavaScript notation

```
var myJSONObject = {"bindings":
 [
  {"ircEvent": "PRIVMSG", "method": "newURI", "regex": "^http://.*"},
  {"ircEvent": "PRIVMSG", "method": "deleteURI", "regex": "^delete.*"},
  {"ircEvent": "PRIVMSG", "method": "randomURI", "regex": "^random.*"}
 ]
};
```

```
myJSONObject.bindings[0].method  // "newURI"
```

Parsing JSON

- ## In Java
  - – Freely available parsing class

Method Summary		
JSONObject	**accumulate**(java.lang.String key, java.lang.Object value) Accumulate values under a key.	
JSONObject	**append**(java.lang.String key, java.lang.Object value) Append values to the array under a key.	
static java.lang.String	**doubleToString**(double d) Produce a string from a double.	
java.lang.Object	**get**(java.lang.String key) Get the value object associated with a key.	
boolean	**getBoolean**(java.lang.String key) Get the boolean value associated with a key.	
double	**getDouble**(java.lang.String key) Get the double value associated with a key.	
int	**getInt**(java.lang.String key) Get the int value associated with a key.	
JSONArray	**getJSONArray**(java.lang.String key) Get the JSONArray value associated with a key.	

Summary

- **Web services**
  - Access to remote procedures / data
  - Promotes integration
    - Better than everybody inventing custom interchange schemes
  - Makes it through firewalls
  - Runs on top of the mature web architecture
- **Common data formats:**
  - XML
  - JSON
- **Possible paper / presentation topics:**
  - Analysis of differences between SOAP and REST
  - Learn details and do something interesting with a specific API
    - e.g. What can be done with the Facebook API?