

An Introduction to Data Publishing and RESTful APIs

Bill Rensmith, Windsor Solutions, Inc.

EN2014 - 2/25/2014

What is an API?

- **API: Application Programming Interface**
 - The mechanism in which a program allows other programs to interact with it



```
Option Explicit
Private Sub Button1_Click()
    Dim Counter As Double
    Dim FirstColumn As Double, LastColumn As Double
    Dim i As Double
    Const delay As Double = 0.825 ' seconds
    Const step As Double = 0.82

    FirstColumn = ActiveSheet.Range("FirstColumn").Value
    LastColumn = ActiveSheet.Range("LastColumn").Value

    For Counter = FirstColumn To LastColumn Step step
        ActiveSheet.Range("ControlColumn").Value = Round(Counter, 1)
        DoEvents
        i = Timer
        Do
            If Timer - i > delay Then Exit Do
            DoEvents
        Loop
    Next Counter
End Sub
```

Application Programming Interface



Human Interface

Using the MS Word API

Demo:

Using the Word API, create a new Word document and insert some text from a different program

The Web Has APIs too!



The screenshot shows a web browser window with the address bar containing the URL <https://mathieu.fenniak.net/2013-most-popular-and-successful-web-apis/>. The page title is "Mathieu Fenniak". Below the title is a large image of a server rack with blue and yellow cables, and a laptop on the floor. The navigation menu includes "HOME", "ABOUT", and "APISLICE: API DOCUMENTATION VERIFICATION". The main content area features the article title "2013 Most Popular and Successful Web APIs" in blue. The article text begins with "Have you ever been stumped how to design a Web API? There are so many different approaches that you can take for even simple things, like paginating API result sets. If you list out your options, it can be easy to end up in analysis paralysis." and continues with "I like to remind myself that I'm not the first person to tread on this soil. I often take inspiration on design choices from well-known APIs, assuming that the hard work has already been done. But I realized lately that I'm not confident what a popular or successful API is, and that I have my own exposure biases in what is popular or successful. So, I decided to do some research to address that."

03/06/2013

0 comment

Tweet 16

Like 6

2013 Most Popular and Successful Web APIs

Have you ever been stumped how to design a Web API? There are so many different approaches that you can take for even simple things, like paginating API result sets. If you list out your options, it can be easy to end up in analysis paralysis.

I like to remind myself that I'm not the first person to tread on this soil. I often take inspiration on design choices from well-known APIs, assuming that the hard work has already been done. But I realized lately that I'm not confident what a popular or successful API is, and that I have my own exposure biases in what is popular or successful. So, I decided to do some research to address that.

What is a Web API?

- A way for programs to access information and functionality over the Internet using basic web communication standards
- Allows for integrating services and information from one system into another
- Documentation allows a developer to learn how to write programs that interact with the API

An API Comparison

Microsoft Word API

Netflix Web API



Word
Engine



Human Interface

API Interface to
other Programs



Netflix
Engine

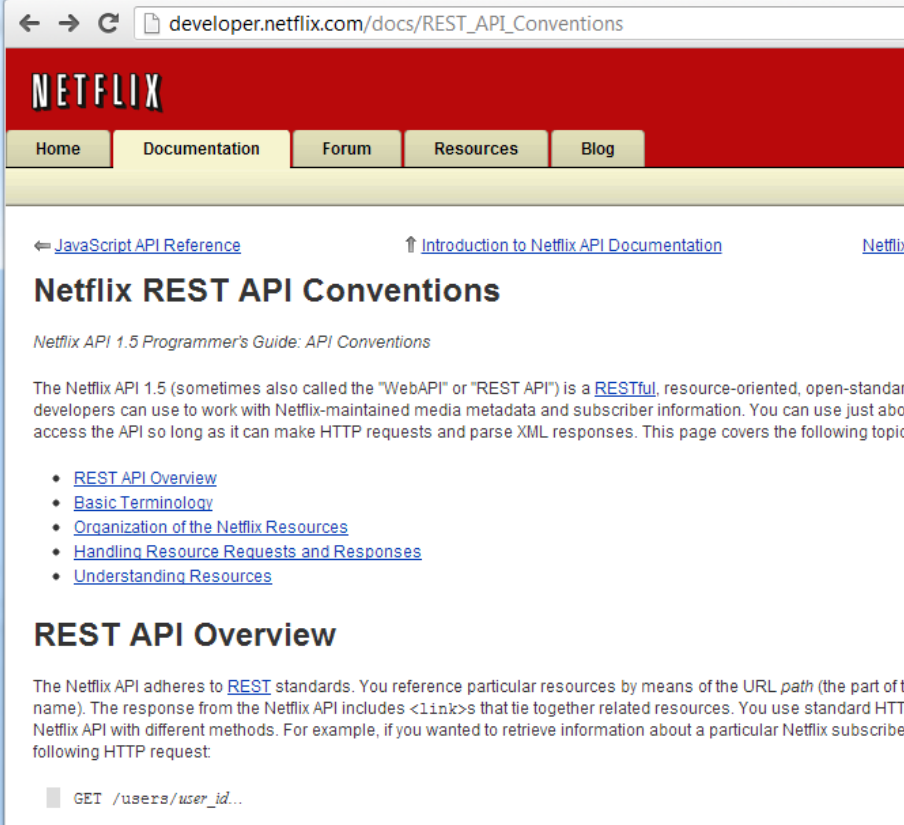


Human Interface

API Interface to
other Web Apps

Simple Web API Example - Netflix

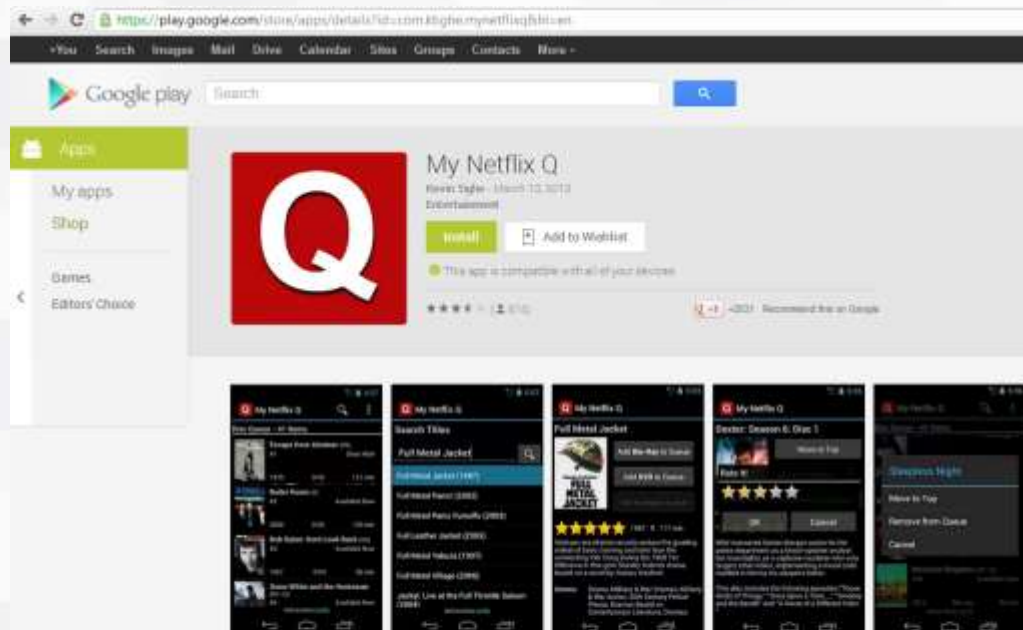
- Search Netflix Catalog
 - By Title
 - By Cast/Director Name
- Get Title Details
 - Synopsis, Release Year, Cast, Director, Rating...
- Get Similar Titles
- View Instant Queue
- Update Instant Queue



The screenshot shows a web browser window with the URL `developer.netflix.com/docs/REST_API_Conventions`. The page features the Netflix logo at the top, followed by a navigation menu with links for Home, Documentation, Forum, Resources, and Blog. Below the navigation, there are breadcrumb links: `← JavaScript API Reference`, `↑ Introduction to Netflix API Documentation`, and `Netflix`. The main heading is **Netflix REST API Conventions**, with a subtitle *Netflix API 1.5 Programmer's Guide: API Conventions*. The text explains that the Netflix API 1.5 is a RESTful, resource-oriented API used for media metadata and subscriber information. A list of links includes [REST API Overview](#), [Basic Terminology](#), [Organization of the Netflix Resources](#), [Handling Resource Requests and Responses](#), and [Understanding Resources](#). Below this is a section titled **REST API Overview**, which states that the API adheres to REST standards and provides an example of a GET request to `/users/user_id...`.

What good is a Netflix API?

- Wouldn't it be cool to quickly add a movie to your Netflix Instant Queue from your smart phone? There's an App for that!



Google play

My Netflix Q
Rivers Taylor · March 13, 2013
Entertainment

Watch Add to Wishlist

This app is compatible with all of your devices

★★★★☆ 110

Recommended by Google

Description

A simple app to manage your netflix queues

Features

- Ability to manage disc and instant queues
- Search for titles
- View netflix recommendations
- Browse titles by category
- Rate titles
- No ads

How Do Web APIs work?

- Most everything that happens on the web rides on the **HTTP** highway
 - HTTP is a set of transport protocols and rules
 - Everything is a Request/Response pair (round trip)
 - Client makes the **request**
 - Server provides a **response**
- HTTP is **synchronous**
 - synchronous means the connection stays open until complete response is received by the client

How Do Web APIs work? (cont'd)

- HTTP is very flexible in the payload it carries
 - **Request** can be just a simple URL or a whole form of data you are submitting (or other things)
 - **Response** could be a web page, a file (zip, pdf, etc.), streaming media, or an HTTP Status Code
 - Ever get an “HTTP 404 – Not Found” message?
- Web APIs use HTTP as the carrier, just like you do when you are surfing the web

HTTP Status Code 418 – I am a Teapot



418
I'm a teapot

Are There Different Types of Web APIs?

- **SOAP**
 - The Exchange Network standard
- **REST**
 - A simpler approach than SOAP
- **Atom/RSS**
 - mainly for publishing/reading news feeds
- **Others**
 - not significant enough to mention
- We will focus on SOAP and REST

What is REST?

- Representational State Transfer (REST) is a software architectural style
- REST is neither a specification nor a technology
- It is a design approach whereby any resource on the Internet can be REPRESENTED by a URL
- Builds naturally upon HTTP
- Requires no special toolkits, frameworks or other overhead for programmers or applications to deal with

(Fictitious) RESTful URL Examples

<http://waterdata.gov/GetSamples?CharacteristicName=zinc>

<http://brownfields.org/Site/1234>

Demo 1 – Simple RESTful Query


- List Beach Advisories in Michigan
 - <http://www.deq.state.mi.us/beach/rest/advisory.aspx>
 - No Criteria on URL
- List Michigan Beaches, filtered by Beach Name
 - <http://www.deq.state.mi.us/beach/rest/beach.aspx?beachtype=Public&waterbodybeach=Harrisville>
- Services are used by myBeachCast app in Google Play store



APIs are for Reading and/or Writing

- **Reading:** Getting information from Server without changing anything on the server
 - Example: Search Netflix Titles
- **Writing:** Interacting with the Server in a way that changes the information on the server
 - Example: Updating Netflix Instant Queue

Reading vs. Writing On the Web

- **Reading** data is typically done with a [hyperlink](#)
 - HTTP “GET”
 - Only thing in the request is the URL
- **Writing** is typically done with a button 
 - HTTP “POST”
 - Request is a URL and all the form data is in the request body
- **GET** and **POST** are “HTTP Verbs”

You never see them in your browser, but the verb is sent behind the scenes

RESTful URL Examples Revisited

<http://waterdata.gov/GetSamples?CharacteristicName=zinc>

- Reading only, as implied by URL text “GetSamples”

<http://brownfields.org/Site/1234>

- Retrieve site 1234 using HTTP Verb **GET**
- Update site 1234 using HTTP Verb **POST**
- Delete site 1234 using HTTP Verb **DELETE**

This is what we mean when we say
REST “builds naturally on HTTP”

So is the Exchange Network a Web API???

YES

How Does the Exchange Network Fit In?

- Exchange Network uses **SOAP** syntax
- Modern Web APIs use **REST** syntax
- Some Similarities
 - Both are carried over HTTP or HTTPS
 - Both are capable of reading/writing data
 - Both support user authentication
 - EN SOAP *requires* it (a policy issue), REST can support it if needed

Exchange Network Parallels

- **Reading**
 - Query
 - Solicit
 - Get Status
 - Download
- **Writing**
 - Submit
 - Execute

Comparing SOAP vs REST Syntax

- SOAP is much more verbose and complex

SOAP 'Authenticate' Request	REST 'Authenticate' Request
<code>https://www.myagency.gov/node/Endpoint2/ENService20.asmx</code>	<code>https://mysite.com/Authenticate?userid=cdx&credential=test&authenticationMethod=password</code>
HTTP "POST"	HTTP "GET"
<pre><s:Envelope> <s:Body> <Authenticate> <userId>cdx</userId> <credential>test</credential> <domain>default</domain> <authenticationMethod> password</authenticationMethod> </Authenticate> </s:Body> </s:Envelope></pre>	none!

SOAP Strengths

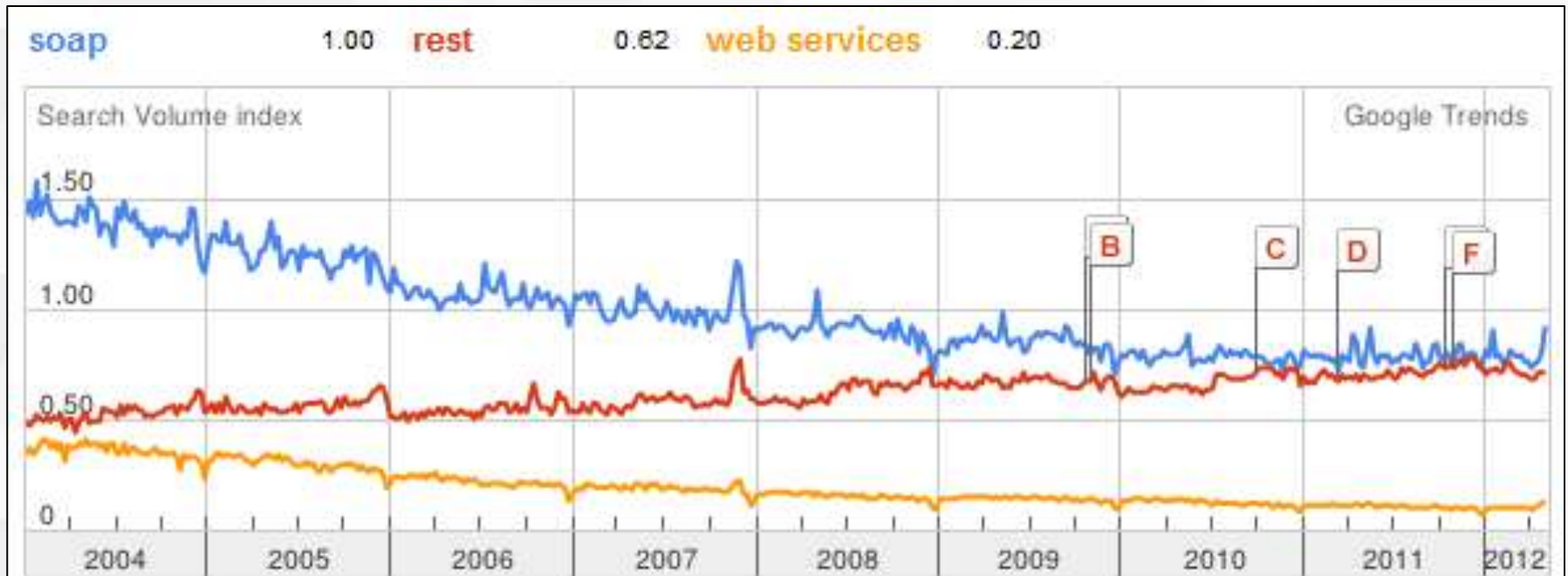
- Uses formally defined and adopted protocols
- Supported by all major programming frameworks
- Ideal for asynchronous operations
 - think standard EN Submit/GetStatus/Download workflow
- Ideal for very large messages
- Enforces a formal contract between sender and receiver
 - Ideal when many partners need a common way of communicating
- The first and most formal web API standard

REST Strengths

- Really simple to publish data for both data provider and data consumer
- Easily integrated into a variety of devices and platforms (think mobile apps)
 - Lots of developer tools
- Widespread adoption and acceptance in developer community
- Lightweight and efficient
- Underlying technology for web “mashups”

The Rise of REST

- REST has really taken off, supplanting SOAP for the preferred Web API Technology



Why is REST Gaining Popularity?

- IBM Whitepaper, Nov. 2008:
“...(REST) has gained widespread acceptance across the Web as a **simpler alternative** to SOAP-based Web services. Key evidence of this shift in interface design is the adoption of REST by mainstream Web 2.0 service providers—including Yahoo, Google, and Facebook—who have deprecated or passed on SOAP and WSDL-based interfaces in favor of an **easier-to-use**, resource-oriented model to expose their services”

<http://www.ibm.com/developerworks/webservices/library/ws-restful/>

Retrieval Formats:

SOAP is XML / REST is More Flexible

- Differences (continued)
 - **EN SOAP** always returns an **XML Document**
 - **REST** can return any variety of formats:
 - XML
 - HTML
 - JSON
 - PDF
 - ZIP
 - ...you name it!

Demo #2 – Different Result Formats

- <http://www.waterqualitydata.us/>
 - Retrieves WQX and other surface water quality data
 - **Inputs:** state, county, lat/long box, characteristic name, etc.
 - **Outputs:** List of Sites OR List of Sites and Results
 - Demonstrates various response formats



The screenshot shows the National Water Quality Monitoring Council (NWQMC) Water Quality Portal (WQP) search interface. The page features a header with the NWQMC logo and the text "NATIONAL WATER QUALITY MONITORING COUNCIL" and "Water Quality Portal". Below the header is a navigation bar with links for "WQP Home", "Download Data", "How to use the WQP", "National Results Coverage", and "About the WQP". The main content area is divided into several sections for data entry:

- LOCATION:** Includes fields for Country (US), State (US 50), and County (US 31000).
- Point location:** Includes fields for Address, City/State, and Zip/Postal.
- Recording Area:** Includes fields for North, East, and South.
- WELL PARAMETERS:** Includes fields for Well Type, Organization ID, Site ID, and HUC.
- SAMPLING PARAMETERS:** Includes fields for Sample Name, Characteristic Group, Characteristic, Parameter Code (with a dropdown menu), and Date Range (From and To).

Hold on, what's this JSON thing?

- Just another data format, like XML only has different syntax
- Popular because it is easily manipulated in web page code

XML

<Permit>

<IssueDate>12/1/2004</IssueDate>

<EffectiveDate>1/1/2005</EffectiveDate>

<ExpirationDate>12/31/2010</ExpirationDate>

<PermitteeName>ACME Inc.</PermitteeName>

<IsInCompliance>True</IsInCompliance>

<Permit>

JSON Equivalent:

```
{“IssueDate”:”12/1/2004”, “EffectiveDate”:”1/1/2005”, “ExpirationDate”:”12/31/2010”, “PermitteeName”:”ACME Inc.”, “IsInCompliance”:”true”}
```

JSON Syntax Rules

- **Name/Value pairs** separated by a colon
 - Name1:Value1
 - Example: “PermitteeName”:”ACME Inc.”
- **Object**: wrapped in braces. Comma separates name/value pairs
 - {name1:value1,name2:value2}
- **Array** (A list of objects): wrapped in brackets
 - [{name1:value1,name2:value2}{name1:value1,name2:value2}]

How is the EN SOAP Different from REST? (cont'd)

- Service Format
 - **EN SOAP** – Every service **must strictly conform** to the EN service standard (WSDL)
 - For example Query must have the following parameters:
securityToken, dataflow, request, rowId, maxRows, parameters
 - **REST** – API developer decides how to construct the service
 - Parameters to support
 - Response formats to support
 - Any aspect of behavior is at behest of developer
 - EN guarantees **consistency**, REST guarantees **flexibility**

How is the EN SOAP Different from REST? (cont'd)

- Service Documentation
 - EN SOAP
 - Every flow has a vetted, published FCD, DET, and schema published to www.exchangenetwork.net
 - REST
 - API developer publishes a web page on how to interact with the service, including examples
 - Can register with online registry (programmableweb.com)
 - Again, EN guarantees **consistency**, REST guarantees **flexibility**

So Is REST or SOAP Better?

- Neither is “better”. They are good at different things.

SOAP	REST
Standards-based	No official standard
Ideal for asynchronous processes (submit)	Ideal for synchronous processes (query)
Slower	Faster
Rigid and Specific	Flexible and Fluid
Ideal for complex business processes/systems	Ideal for mobile devices
Difficult to build client apps	Simple to build client apps

REST Support on the EN Today

- REST Guidance document published May, 2013
 - <http://www.exchangenetwork.net/rest-guidance/>
 - Only covers Query services
 - Describes standards for parameter encoding in URL and documentation, and registering in ENDS
- CDX REST Proxy
 - Described in REST Guidance document, Section 6

Operations at <https://ends2.epacdxnode.net/RestProxy>

This page describes the service operations at this endpoint.

Uri	Method	Description
Authenticate	GET	Service at https://ends2.epacdxnode.net/RestProxy/Authenticate?Id={USERID}&Credential={CREDENTIAL}&Domain={DOMAIN}&Method={METHOD}
Query	GET	Service at https://ends2.epacdxnode.net/RestProxy/Query?Node={NODE}&Dataflow={DATAFLOW}&Request={REQUEST}&Params={PARAMETERS}&RowId={ROWID}&MaxRows={MAXROWS}&Token={TOKEN}

REST Support on the EN Today (cont'd)

- **All Node Products incorporate REST Support today**
 - EN-Node
 - OpenNode2
 - EPA Virtual Node
- **EPA REST Services**
 - RCRA and WQX offer GET RESTful services
 - ...numerous others
- **Institutional Controls FCD has RESTful URL Examples**



The Future of REST on the Exchange Network

Kurt Rakouskas