REST based API

Shabnam Kumari
A.P, Department of CSE,
Sat Kabir Institute of Technology & Management,
Bahadurgarh, Haryana, India

Deepak
M.Tech Scholar, Department of CSE, Sat Kabir Institute of Technology & Management,
Bahadurgarh, Haryana, India

ABSTRACT

Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) are two answers to the same question: how to access Web services. The choice initially may seem easy, but at times it can be surprisingly difficult. SOAP is a standards-based Web services access protocol that has been around for a while and enjoys all of the benefits of long-term use. Originally developed by Microsoft, SOAP really isn't as simple as the acronym would suggest. The Difference between SOAP vs REST APIs REST is the newcomer to the block. It seeks to fix the problems with SOAP and provide a truly simple method of accessing Web services.

Keywords: SOAP, REST, API

1. INTRODUCTION

RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

REST stands for REpresentational State Transfer. REST is a web standards based architecture and uses HTTP Protocol for data communication. It revolves around resources where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in year 2000.

In REST architecture, a REST Server simply provides access to resources and the REST client accesses and presents the resources. Here each resource is identified by URIs/ Global IDs. REST uses various representations to represent a resource like Text, JSON and XML. JSON is now the most popular format being used in Web Services.

1.1 HTTP Methods

The following HTTP methods are most commonly used in a REST based architecture.

- GET − Provides a read only access to a resource.
- PUT − Used to create a new resource.
- DELETE − Used to remove a resource.
- POST − Used to update an existing resource or create a new resource.
- OPTIONS − Used to get the supported operations on a resource.

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

Web services based on REST Architecture are known as RESTful Web Services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI (Uniform Resource Identifier), which is a service that provides resource representation such as JSON and a set of HTTP Methods.

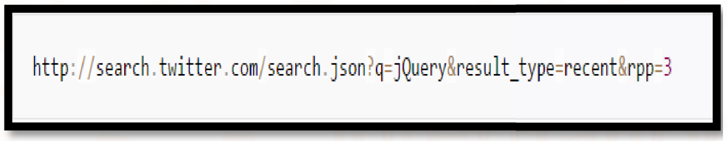
2. Web Service API examples:

Because REST API's use HTTP, they can be used by practically any programming language and easy to test (it's a requirement of a REST API that the client and server are independent of each other allowing either to be coded in any language and improved upon supporting longevity and evolution).

The World Wide Web (WWW) is an example of a distributed system that uses REST protocol architecture to provide a hypermedia driven interface for websites. I'm saying hypermedia (instead of hypertext) as an expansion term to avoid confusion about the REST API supporting other formats to be provided not just HTML.

2.1. Real World Examples:

- Twitter API :
  Twitter provides a REST API which you can query to get the latest tweets, you can provide a search query (or hash tag) and it will return the results in JSON format. Example of this HTTP request to the Twitter API to get the latest 3 tweets matching "jQuery".



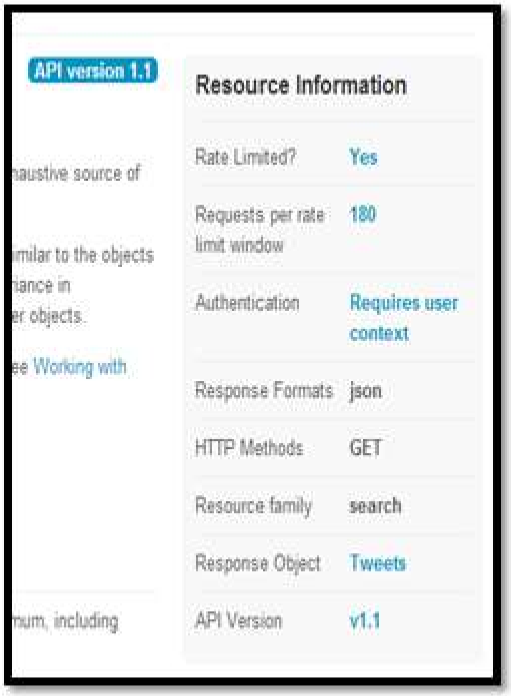
```
http://search.twitter.com/search.json?q=jQuery&result_type=recent&rpp=3
```

Figure 2.1. Twitter API URL

And to expand on what a REST API should provide:

The REST API should specify what it can provide and how to use it, details such as query parameters, response format, request limitations, public use/API keys, method (GET/POST/PUT/DELETE), language support, callback usage, HTTPS support and resource representations should be self-descriptive.

This is the information provided for the GET search/tweets REST API.



API version 1.1

haustive source of

imilar to the objects
hance in
er objects.

ee Working with

mum, including

Resource Information

Rate Limited?	Yes
Requests per rate limit window	180
Authentication	Requires user context
Response Formats	json
HTTP Methods	GET
Resource family	search
Response Object	Tweets
API Version	v1.1

Figure 2.2. Twitter API Resource Information

2.2. API Providers :

Following are the other major API providers:

- The Twitter REST API
- Facebook REST API
- Google Translate REST API
- Flickr REST API
- Dropbox REST API
- Ebay Developer REST API (product centres)
- BING Maps REST API (services)
- BING Traffic Incidents API
- Magento REST API

3. Login API Integration Code:

In our project, we are integrating APIs from social networking sites to allow users to register and login to our website and we will also fetch the content from those social networking sites.

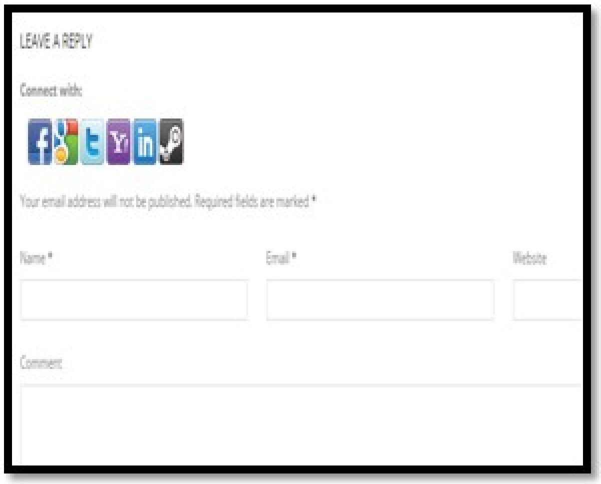
Below is the main code for providing OAuth to pass to the API calls for social sites like Facebook, Instagram, Twitter, and Google Plus.

```
class Hybrid_Auth

{

        public static $version = "2.4.1-wsl-fork";

        public static $config  = array();

        public static $store   = NULL;

        public static $error   = NULL;

        // ----------------------------------------------------

        /**

        * Try to start a new session of none then initialize Hybrid_Auth

        *

        * Hybrid_Auth constructor will require either a valid config array or

        * a path for a configuration file as parameter.

        */

        function __construct( $config )

        {

                Hybrid_Auth::initialize( $config );

        }

        // ------------------------------------------------------

        /**

        * Try to initialize Hybrid_Auth with given $config hash or file

        */

        public static function initialize( $config )

        {

                if( ! is_array( $config ) && ! file_exists( $config ) ){

                        throw new Exception( "Hybriauth config does not exist on the given path.", 1 );

}
```

4. Results:

4.1. API Integration Output:

Below are some screenshots of the project code when implemented with a valid API keys required for the target websites.



LEAVE A REPLY

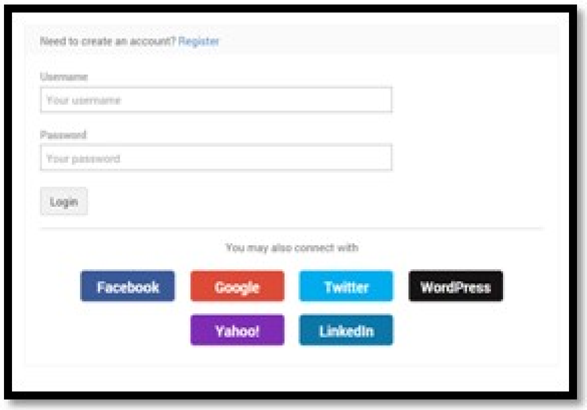
Connect with:

Your email address will not be published. Required fields are marked *

Name * Email * Website

Comment

Figure 4.1 Output Connect with social sites



Need to create an account? Register

Username

Your username

Password

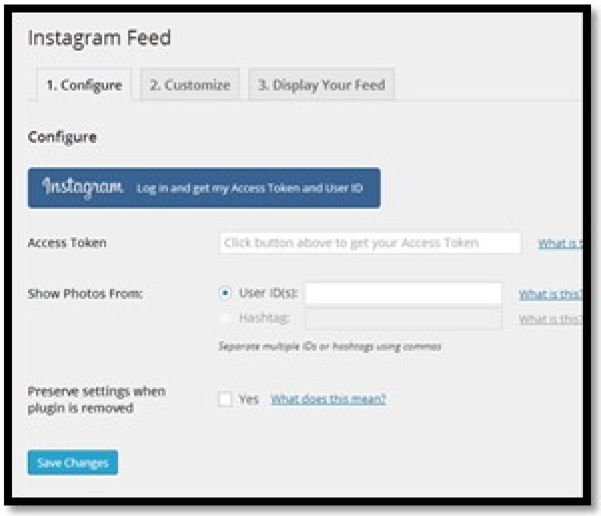
Your password

Login

You may also connect with

Facebook Google Twitter WordPress Yahoo! LinkedIn

Figure 4.2. Output login with social sites



Instagram Feed

1. Configure 2. Customize 3. Display Your Feed

Configure

Instagram Log in and get my Access Token and User ID

Access Token: Click button above to get your Access Token

Show Photos From: [✓ User ID(s):] Hashtag:

Separate multiple IDs or hashtags using commas

Preserve settings when plugin is removed: Yes What does this mean?

Save Changes

Figure 4.3. Output Instagram token settings

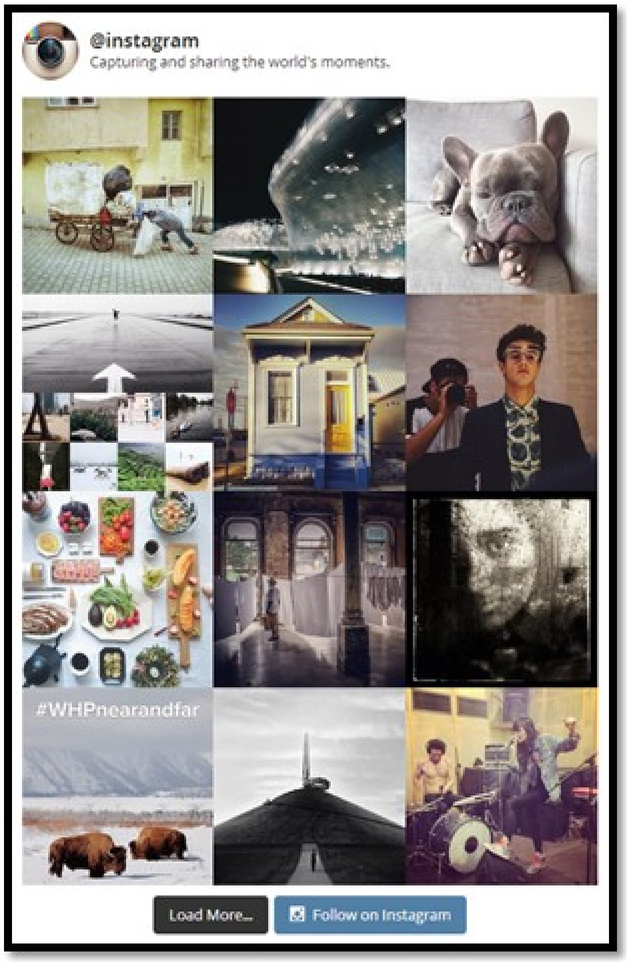


Figure 4.4. Output Instagram Images

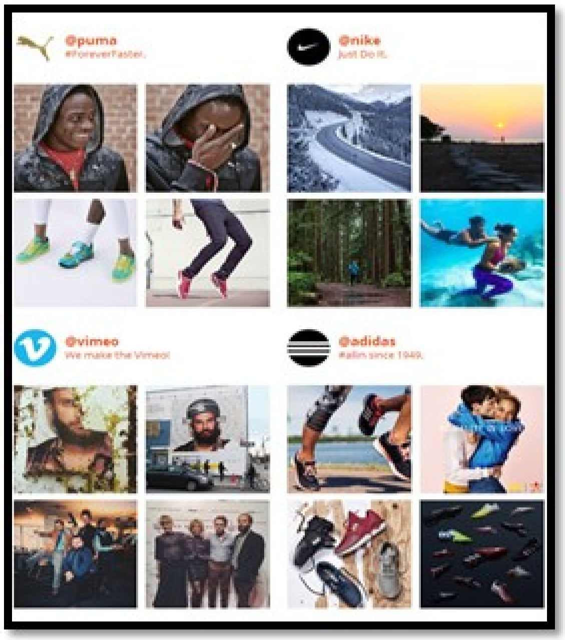
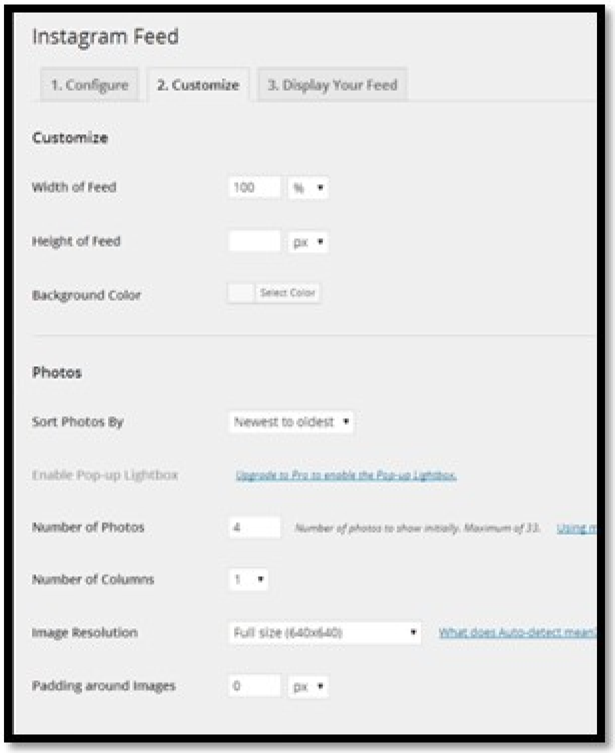


Figure 4.5. Output Instagram Images layout 1



Instagram Feed

1. Configure | 2. Customize | 3. Display Your Feed

Customize

Width of Feed	100 %
Height of Feed	px
Background Color	Select Color

Photos

Sort Photos By	Newest to oldest
Enable Pop-up Lightbox	Upgrade to Pro to enable the Pop-up Lightbox.
Number of Photos	4 Number of photos to show initially. Maximum of 33.
Number of Columns	1
Image Resolution	Full size (640x640) What does Auto-detect mean?
Padding around Images	0 px

Figure 4.6. Output Instagram image settings

Figure 4.7. Output Instagram Images layout 2

5. Conclusion:

Continuing our discussion of using the web services layer as a pure packaging strategy of a business module, we evolved the architecture further to discuss message interfaces. We've explored XML, SOAP, REST and WSDL web services and identified issues with WSDL as the interface definition. Using the XSD to define the message interfaces provides separation of responsibility between the business module and the web services infrastructure layers. Further, it allows the enterprise to have its business module interfaces defined and managed by its business groups, where such responsibility belongs. The engineering group is responsible for the implementation of the business functionality and can focus on its task with clear business functionality definitions already provided. Web services and XML technologies are evolving enterprise processes to truly define ownerships and responsibility within groups—without any constraints of technology, platform, or products used within the enterprise.

Web services are one of the key elements of the so-called programmable Web. They are extremely versatile software elements that really have the potential to open up a new era in software: the age of interoperability. Web services can be effectively used to participate in and set up business-to-business (B2B) transactions. They are great at exposing software functionality to customers and integrating heterogeneous platforms.

References:

[1] http://en.wikipedia.org/wiki/SOAP

[2] http://en.wikipedia.org/wiki/Representational_state_transfer

[3] http://en.wikipedia.org/wiki/Web_service [4] http://stackoverflow.com/questions/ 19884295/soa p-vs-rest-differences

[5] http://stackoverflow.com/questions/4163066/restvs-soap-has-rest-a-better-performance

[6] http://blog.smartbear.com/apis/understandingsoap-and-rest-basics/

[7] REST vs. SOAP: Making the Right Architectural Decision by CesarePautasso from Faculty of Informatics University of Lugano(USI), Switzerland;

[8] Sharing Service Semantics using SOAP-Based and REST Web Services by Xuan Shi

[9] SOAP and Web Services by Panagiotis Louridas

[10] SOAP-Based vs. RESTful Web Services by Fatna Belqasmi, Jagdeep Singh, Suhib Younis Bani Melhem, and Roch H. Glitho from Concordia University

[11] T. Berners-Lee, R. Fielding, and H. Frystyk. RFC 1945: Hypertext Transfer Protocol — HTTP/1.0, May 1996.

[12] J. Correia and M. Cantara. Gartner sheds light on developer opps in web services. Integration Developers News, June 2003.

[13] Dare Obsanjo Blog. Misunderstanding REST: A look at the Bloglines, del.icio.us and Flickr APIs, May 2011. http://www.25hoursaday.com/weblog/PermaLink.aspx?guid=7a2f3df2-83f7-471bbbe6-2d8462060263.

[14] B. M. Duska, D. Marwood, and M. J. Freeley. The measured access characteristics of World-WideWeb client proxy caches. In USENIX Symposium on Internet Technologies and Systems, USITS, 1997.