

12-2013

# TEACHING WEB APIS IN INTRODUCTORY AND PROGRAMMING CLASSES: WHY AND HOW

Timothy Olsen

Kathleen Moser

Follow this and additional works at: <http://aisel.aisnet.org/siged2013>

---

## Recommended Citation

Olsen, Timothy and Moser, Kathleen, "TEACHING WEB APIS IN INTRODUCTORY AND PROGRAMMING CLASSES: WHY AND HOW" (2013). *2013 Proceedings*. 16.  
<http://aisel.aisnet.org/siged2013/16>

This material is brought to you by the SIGED: IAIM Conference at AIS Electronic Library (AISeL). It has been accepted for inclusion in 2013 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

## TEACHING WEB APIS IN INTRODUCTORY AND PROGRAMMING CLASSES: WHY AND HOW

Timothy Olsen  
Department of Information Systems  
Arizona State University  
Tim.Olsen@asu.edu

Kathleen Moser  
Department of Information Systems  
Arizona State University

### Abstract:

Web services technology has received much attention in recent years under a broader umbrella of service-oriented architecture (SOA). In a memo to top-managers at Microsoft, Bill Gates stated “The broad and rich foundation of the internet will unleash a ‘services wave’ of applications and experience available instantly over the internet to millions of users” [Gates, 2008]. While Web Services are widely used in developing systems of today, they are often excluded from information systems curriculum (Lim et al., 2010; Lopez et al., 2007). We describe how and why teaching modern web services is an important part of information systems curriculum, and how they can be introduced at introductory levels. Practical suggestions are then offered for other instructors to integrate Web Services as part of a rich classroom experience, outlining the broad benefits to students that will better prepare them for upper division classes or to add to their skill set for industry positions upon graduation.

**Keywords:** web api, REST API, evented web, service-oriented computing, teaching APIs

### I. Introduction

Over the past five years, enterprises have moved existing applications into cloud service providers in order to save on hardware and data-center costs. As small and large businesses adopt cloud-based applications, they have begun to deal with the complexities of working with distributed systems, and distributed data. Web application programming interfaces (APIs) have emerged as the standard in integrating data and events between the enterprise and this distributed cloud-based environment. An API is a way for two computer applications to talk to each other over a network using a common language. Essentially, they are a way to share data between business partners and applications (Apigee 2011).

Web APIs are similar to but distinct from other web service protocols. Most notably Web Services Description Language (WSDL), which describes the web service and Simple Object Access Protocol (SOAP) which explains the communication protocols for exchanging information over computer networks. These web service protocols provide a description of how the service can be called, what parameters it expects, and what data structures it returns and then offers the services over the internet.

The growth of web services and Web APIs marks a fundamental change in building systems. One way of conceptualizing this change is a move from modularity to generativity [Yoo, 2012]. Web APIs are often built with the intention that they will be used to build products and services not imagined by those who created the API. For example, the Google maps service can be integrated into mobile phones, TVs, navigation systems, real-estate listing services and other services that Google may not have originally intended. Whereas modular products begin with fixed boundaries, generative products (Web APIs) are designed without fully knowing the “whole” of how each module will be integrated with others [Gawer, 2009; Yoo, 2012].

Web APIs are also known as Hypertext Transfer Protocol (HTTP) APIs for message negotiation and transmission that work well with network firewalls or as Representational State Transfer (RESTful) Web Services where requests and responses are built around the transfer of concept states and hypermedia APIs.

With the rise of cloud-based services Web APIs have easily eclipsed the number of WSDL/SOAP web services in a few short years. Table 1 [Adamczyk et al., 2011] illustrates the growth of these web services and a projection to 2013, based on data collected from programmableweb.com. Some services are available in two or more styles; the number of unique services is shown. Table 2 (Iyer 2013) illustrates the history of general API use by traditional businesses.

Table 1: Web Service Styles Used in Public Services

Web Service Growth			
Style	2007	2010	2013
Web APIs	180	1627	6371
SOAP	101	368	2052
XML-RPC	24	53	194
Other	90	207	624
Total	395	2255	9241
Unique	340	2179	9888

Table 2. History of API use

1990s	Operating Systems Vendors (IBM, Microsoft, Red Hat, etc.)
2000s	Internet-centric companies expose information assets (eBay, Amazon, Netflix)
2010s	Traditional Brands (Mastercard, Comcast, USA Today, New York Times, Best Buy, National Public Radio, and others)

It is clear that the use and number of Web APIs is growing and will continue to grow significantly and have become fundamental in systems development, however they are often not taught in information systems courses. In our experience, teaching Web APIs have increased student engagement in the classroom in several different platforms (graduate, undergraduate) and in different types of courses (survey, process management, systems analysis). In the next section some of the benefits of adding Web APIs that we've experienced are described.

## II. WHY TEACH WEB APIS?

There are many reasons for including an element of Web APIs into introductory information systems or programming courses. We have experience teaching API concepts to MBA, Masters of IS, and undergraduate students at senior levels. In all cases, students were delighted and expressed amazement of how easy it was to share information across parties and systems.

Our experience teaching this includes:

- Two days as part of a “current topics” module in a senior capstone course
- An hour discussion at the Master’s level (business process course)

- An entire semester course based on the tools outlined in the “Incorporation into Existing Systems” section below.
- For the IS intro class we suggest two days on this topic

From our experience we outline some of the benefits we have seen.

### **Increased Enrollments**

Since the dot-com bubble burst universities have faced a decline in the number of students who study in technical disciplines such as information systems, or computer science. As a response to this universities have modified or marketed their programs to increase the number of students. In addition, the NSF and other bodies have issued solicitations to fund research and innovative advances to improve curriculum with the intent of boosting enrollment [Lim et al., 2010; Olsen and Welke, 2008].

### **Enjoyable for Students**

Introducing Web APIs in introductory information systems classes has increased student excitement for the information systems majors. We received very high student ratings when this subject was introduced. We have also observed that advanced and experienced senior students are excited to create new systems that utilize and combine several cloud-based services. A sampling of comments from students who shared what they learned in the class, how it changed their perspective, and the high interest level are quoted below.

- "What I really liked was the yelp assignment we did in class, where we got our first taste of using connectors and finding the API and how to link it to our process. With that we completed multiple tasks in one process where we found a phone number, neighborhood, and category of a specific item such as food. This allowed us to narrow down the information, choose the information and use another application, Twilio, to send out the information to yourself or a friend."
- "Prior to this course, I had little to no understanding of what an API was and what could be created from them. Now I am confidently and actively searching for services that can be provided by combining different technologies."

### **Adds Valuable Skills for Advanced Courses and Industry Positions**

Being able to use an API will become increasingly important in classes and work. As businesses continue to adopt cloud-based services, the ability to share data via APIs will become more important. Not only will this be a crucial skill for IT positions, as evidenced by the quote below, it is also useful for managers.

"Monitoring the variables, design, and functionality were all skills I learned in my basic programming classes, however were reinforced and even more vital when combining multiple web services together."

### **Useful for Non-programmers**

The existence of Web APIs represents a huge advantage not only to programmers, but also to management. We have found that even non-technical managers are able to utilize knowledge of Web APIs to integrate services, and automate simple office tasks or workflows. Students who will not be programming full-time upon graduation can still fully appreciate the benefits from this knowledge.

"I learned to automate processes, for example, in the Google Site Comment assignment. We were required to use the Google API login and Google sites pre-configured web services to launch a successful process for a user to add

a comment to a Google site via an interface. Integrating the web services in this business process example was an easier and quicker method for non-coders to create processes."

### III. INTRODUCING WEB APIS INTO THE CURRICULUM

Experience teaching to a diverse audience of undergraduate, professional, and master's students has given us insight on how to teach this in the classroom. In this section, tools and approaches to introducing Web APIs to students at any level (with or without programming experience) is discussed. We modified the approach to be very hands on for the undergraduates, and more conceptual and discussion oriented for the Masters students. We have included our in-class activities as linked documents in the section below.

#### A Basic Introduction for Students

We have found two tools to be useful when introducing Web APIs to students. IFTTT (If This Then That) is a web application that allows you connect actions or events that happen on cloud-based services. If a specific action occurs on one service a user can 'program' another action to occur. For instance if rain is in tomorrows forecast, one can receive a text message the night before. If one is tagged in a photo on Facebook, this photo can be saved to dropbox, as shown in Figure 1. If a certain car has been newly listed for sale on craigslist, an notification email or text-message can be received.



Figure 1. IFTTT

Zapier is similar to IFTTT, but is targeted for enterprises use. It is useful to have students choose between using IFTTT or Zapier during this exercise.

Students are excited to use these services and uncover imaginative uses for them. This can then be used as a dialog of how IFTTT and Zapier systems work leading to a discussion of APIs with the instructor viewing the API documentation for several of the "channels" used by IFTTT. Codecademy provides a very short module on Web APIs that students are able to complete in 15 minutes. This serves as a useful introduction and segues from general awareness to hands-on interaction discussed next.

Resources useful for students/instructors:

- <http://www.ifttt.com>
- <http://www.zapier.com>
- <http://www.codecademy.com/tracks/twilio>

#### Hands-on Interaction

As Web APIs are quite simple, the basic principles are easily learned "hands on". Students will learn that interacting with Web APIs is not terribly different than typing a URL in a

browser address bar. There are several tools which are useful here including: RESTconsole.com, the Poster Firefox plugin, or hurl.it

We chose hurl.it as the most advantageous as it is simple and requires nothing to be installed. During this activity students access a web service and see the result of the service call. Simple services fit for this activity include the use of the Yelp 1.0 API, currency exchange rate or weather APIs. The API that the students seem to enjoy the most at this stage is Twilio, which allows them to send sms messages or phone calls to themselves or others.



Figure 2. Sending a Text Message Using Hurl.it

During this stage students discover the different parts of a Web API service call, as well as hit a few technical errors usually brought on by not having the right access key, or not performing URL encoding.

Resources useful for students/instructors:

- <http://restconsole.com>
- <http://www.hurl.it/>
- <https://addons.mozilla.org/en-us/firefox/addon/poster/>
- <http://home.ccil.org/~cowan/restws.pdf>
- <http://www.slideshare.net/pagsousa/rest-web-services-5478036>
- <http://www.twilio.com/docs/api/rest>
- [http://www.yelp.com/developers/documentation/technical\\_overview](http://www.yelp.com/developers/documentation/technical_overview)
- <http://bit.ly/1ct2peO> - the in-class activity developed by authors as basic introduction

## Incorporation into Existing Systems

After students learn the basics of Web APIs they are able to start implementing them directly into the programming language of their choosing. There are standard libraries for each of the major programming languages that make this easy to accomplish.

Virtually all cloud-based services have Web APIs that work in similar fashion. Our approach has been to allow the students to choose Web APIs to work with, after exposing them to several alternatives. A few use cases that students have created with the tools below include:

- Recreating an emergency alert system using data from Google Spreadsheets API and the Twilio API (call and sms).
- Creating custom invoices for a friends mechanic shop using the Freshbooks API.
- Calling restaurants that meet criteria on Yelp and saving this data for further review.
- Using Amazon Mechanical Turk as labor-as-a-service to write creative stories based on certain genres and time settings.

If students are in a class where learning a programming language is not the objective (business process management, enterprise systems development, systems analysis and design, or an introductory course) they may find the following tools useful to apply their knowledge as they learn to compose systems by bringing together APIs.

*Fujitsu Runmyprocess* – This is a business process management system (BPMS) that allows students to compose a process model that can be integrated with external services to create an application. Unlike other BPMS's this has the advantage of being browser-based, requiring no software installation. After learning the basics of Web APIs, students are easily able to tie them together with a central system to compose a model-based application. We have found this useful for teaching the principles of resource-oriented architecture, business process management, and model-based execution.

*Google Apps Script* works with Google Spreadsheets much like VBA works with Excel. We find that students are able to make significant use of this for several reasons: 1) It is browser-based 2) It uses a very familiar user interface (spreadsheets) 3) Its basis in JavaScript makes connecting with APIs quite easy 4) There is ample online support and documentation.

Resources useful for students/instructors:

- <http://www.Runmyprocess.com>
- <https://developers.google.com/apps-script/>
- <http://bit.ly/1ct2EGA> - in-class activity integrating with Google Apps Script with Twilio REST API to send SMS

## IV. CONCLUSION

Information systems instructors should view introducing the concepts discussed here as “low hanging fruit” which will yield better prepared graduates, more enthused students, and increase the number of students in our programs. Furthermore, these concepts are applicable not only to programming courses, but virtually any information systems class. Knowing the workings of Web APIs is quickly becoming a vital skill of programmers, tech entrepreneurs, and managers of any kind.

## REFERENCES

- Adamczyk, P., Smith, P. H., Johnson, R. E., and Hafiz, M. 2011. "REST and Web Services: In Theory and In Practice," in *The 1st International Workshop on RESTful Design/International Workshop on RESTful Design*, , pp. 35–57.
- Apigee. 2011. *Amundsen's Dogs, Information Halos, and APIs*, Business & Mgmt, , December 1.
- Gates, W. 2008. "Internet Software Services," *memo*, [http://www. scripting. com/disruption/mail. html](http://www.scripting.com/disruption/mail.html), Retrieved May (16).
- Gawer, A. 2009. "Platform dynamics and strategies: from products to services," *Platforms, markets and innovation* , pp. 45–76.
- Iyer, B. 2013. "Bala Iyer," *Time to Be a Business Architect*, .
- Lim, B. L., Hosack, B., and Vogt, P. 2010. "A web service-oriented approach to teaching CS/IS1," in *Proceedings of the 41st ACM technical symposium on Computer science education, SIGCSE '10*, New York, NY, USA, pp. 131–132.
- Lopez, N., Casallas, R., and Villalobos, J. 2007. "Challenges in creating environments for SOA learning," in *Systems Development in SOA Environments, 2007. SDSOA'07: ICSE Workshops 2007. International Workshop on*, , pp. 9–9.
- Olsen, T., and Welke, R. 2008. "Learning Styles as Predictors of Problem-Based Learning satisfaction in the Information Systems Classroom," Presented at the International Academy for Information Management Conference, Paris, France.
- Yoo, Y. 2012. "The Tables Have Turned: How Can the Information Systems Field Contribute to Technology and Innovation Management Research?," *Journal of the Association for Information Systems* (14:5).