{"logo":"API Evangelist"}

# The University of API

Prepared By Kin Lane

August 2014

## Table of Contents

## Why Am I Looking At Universities And APIs

I spend a great deal of time looking at new APIs from companies, institutions, and government agencies during my weekly monitoring of the API space, and over the last couple years, I've come across an increasing number of APIs tht are out of higher educational institutions around the world. It began with a centralized developer area at the University of Washington (UW), and then I saw the same from UC Berkeley, with more recently noticing the impressive conversion of 250+ services from a traditional Service Oriented Architecture (SOA), to a more modern web API approach, at Brigham Young University (BYU).

Along the way I began seeing that APIs at higher education institutions were going to be an important piece of the overall API puzzle, and as I do with any research area, I setup a Github repository (university.apievangelist.com), and began tracking on universities who were doing anything interesting with APIs, no matter how small. I now track on twelve separate higher education institutions that have active API efforts, and I have been engaging in active conversations with two of the institutions, regarding their strategy, working to understand how I can provide any guidance, but also learn from each team about the unique challenges higher education institutions face when designing, deploying, manage, and even evangelize their APIs--this white paper is a result of that ongoing research\

## Something Happening Beyond Campus Information Services

As I was looking through APIs from different universities, I began to notice that campus information technology (IT) were beginning to get organized in how they approached the deployment and management of their APIs, but what really caught my attention was the usage by students as well. While some very progressive IT folks are beginning to understand the benefit of using APIs, and the access and interoperability it can introduce into campus operations, students are also increasingly tech savvy, and are scraping data and content to develop their own API to suppor the apps they build, while also putting pressure on IT to provide more services they can put to use.

All of the activity I was seeing, at a handful of schools, convinced me that I needed to pay closer attention to what universities were up to. I was pleased to also find that half of the universities who were deploying APIs, also had a centralized developer area for their APIs, while the rest of the deployments were singular API efforts, for a specific purpose. I consider the six that have a centralized developer and API strategy, as the leaders when it comes to APIs in higher ed.

# The Leaders When It Comes To APIs In Higher Education

When visiting an API developer area, as soon as the page loads, you can tell if its an organized effort, or something that was thrown up by a single project or team. Half of the institutions I track on as part of my university API monitoring have a centralized developer area for their APIs, as well as what appears to be an organized approach behind the designing, deploying, managing or evangelizing their APIs.

I’ve talked with both University of California Berkeley and Brigham Young Universities about their API strategies, while only having light communication with University of Washington and Michigan. I feel there are some interesting lessons to be had when taking a look at these six leading universities.

## Brigham Young University

Brigham Young University (often referred to as BYU, or sometimes just the Y) is a private university located in Provo, Utah, United States. It is owned and operated by The Church of Jesus Christ of Latter-day Saints (LDS Church), and is the largest religious university and third-largest private university in the U.S.

🏠 https://developer.byu.edu

## University of California Berkeley

The UC Berkeley API Management Portal is your starting place to obtain portable campus data securely. Search and find APIs, learn how to use them, and get the right level of access. Publish your APIs to campus and external audiences, manage security and access levels, and track analytics.

🏠 https://developer.berkeley.edu/

🐦 https://twitter.com/UCBerkeley

## University of Michigan

The University of Michigan, a state university in Ann Arbor, Michigan, offers a variety of APIs that allow developers to access and integrate various data from the University of Michigan with other applications and to create new applications.

🏠 http://developer.it.umich.edu/

🐙 https://github.com/openmichigan

### University of Warsaw

USOS API is a simple and fat-free REST protocol, which allows developers to access academic database. In case of this installation, you will be accessing The University of Warsaw's data, but there are many other institutions which use USOS API.

http://apps.usos.edu.pl/developers/api/

### University of Washington

Web Services at the University of Washington is a method of getting important institutional data from and/or into your applications. Web Services are a way for applications or systems to talk to one another and does not usually involve human interaction. Currently, the use of Web Services is mainly targeted for application developers who would write code against them.

http://webservices.washington.edu/

https://github.com/uw-edu

### University of Waterloo

The University of Waterloo Open Data API allows anyone to build their own programs and applications using data from the University of Waterloo websites. The API features more than 40 methods of accessing various datasets across the University of Waterloo network.

http://api.uwaterloo.ca/

https://uwaterloo.ca/open-data/news

https://uwaterloo.ca/open-data/news/news.xml

https://github.com/uWaterloo

There are some unique aspects to delivering an API program within a higher educational institution, but I'd say 75% is very similar to what occurs in other business sectors. As we do with other industries, as well as the overall API space, we need to showcase these university API pioneers, and model what they are up to, and work to share the stories of success, and even failure, with other institutions. Once the word spreads to other schools, regarding what these six institutions are up to, other schools will being to emulate the patterns, benefiting not just any single school, but also the overall higher education system, and ultimatey the students.

# Building Blocks Used By The Leading Universities

I track on what I consider to be the common building blocks for API design, deployment, management, evangelism, and integration, that can be used in all business, organizational, and government sectors. When it comes to higher education, I’m going to be tracking on a different set of building blocks, showcasing not just the approach, tools, and services used to deploy APIs, I want to focus on which components of campus operations are having APIs applied to them.

While the 250+ APIs at Brigham Young University gave me a huge head start on building this list, I looked across all of the university API leaders, as well as the smaller implementations out there. After aggregating into a single list, I smoothed out some of the rough edges, merged a few of the areas, and organized into this list of common building blocks across university API implementations, as of July 2014.

*(I will add descriptions for each of these in the next edition)*

## Student

**Accounts**

**Alumni**

**Billing**

**Counseling**

**Directory**

Health

Invoices

Lockers

Payments

refunds

Tuition

## Academic

Classes

Credits

Exams

Grades

Records

Registration

Study Abroad

Terms

Transfers

## Faculty

Directory

Research

Tenure

## Courseware

Announcements

Assignents

Courses

Curriculum

Syllabus

## Benefits

Leave

**Retirement**

## Events / Calendaring

**Calendars**

**Incidents**

**Time Away**

## Communications

**Comments**

**Email**

**Messaging**

**Push Notifications**

## File Management

Locations

Photos

Services

## Assets / Inventory

Compute

Dispolsals

Items

Space

## Libraries

Locations

Metadata

Search

## Collections

Corporate

Institution

Journals

Museum

## Organizations

Churches

Forums

Groups

Programs

Transfer Institutions

## Financial Aid

Application

Counseling

Grants

Loans

**Student Information**

## Human Resources

**Career Plans**

**Contacts**

**Employees**

**Endorsements**

**Hiring**

**Positions**

**Training**

## Identity

Access

Organization

Person

## Internship

Applications

Providers

## Payroll

Deductions

Earnings

Payments

Time Reporting

## Purchasing

Change Orders

Purchase Orders

Request for Quotes

Return to Vendors

## Scholarships

Consideration

Directory

Disbursement

## Services

Director

Request

## Vendors

Directory

Payments

Vouchers

## Other

Glossary

News

**Surveys**

These building blocks are meant to be components, that other institutions can consider when planning, deploying and managing their own API efforts. It is not meant to be complete or definitive list, it is meant to act as a guide, allowing IT or even student organizers to better understand how APIs can help universities better deploy web, mobile and single page applications that support campus operations, or possibly be integrated into the classroom.

This list of building blocks will continue to evolve and change as I monitor APIs in higher education, talk to folks who play an active role in these campus API efforts, and learn more about how they view the API design, deployment, and management life cycle. Currently I'm seeing a wide range of motivations for deploying APIs on campus—hopefully with some more work I can better understand these motivations, and share more stories about the problems they solve, to help convince other institutions to follow the lead.

## Making Library Collections Available Via APIs As A Motivation

Beyond the API leaders who have been successful in kicking off a centralized API experience, there are a handful of schools that are only deploying publicly available web APIs as part of their library network. With existing digital catalog, linked data, and other library technology efforts, it make sense that APIs would take root in the university library, within three institutions I'm tracking on.

UNIVERSITY OF CAMBRIDGE
Cambridge University Library

🏠 http://bit.ly/1kjfAnI

### Cambridge

This Cambridge University Library  provides access to the APIs (or Web Services) provided by Cambridge University Library, including XML/JSON based web service interfaces for the library catalogues and other online library services, API stands for 'Application Programming Interface', a mechanism that provides structured information allowing pieces of software to communicate with each other.

HARVARD
LIBRARY

🏠 http://bit.ly/1j56kmm

### Harvard

DASH is Harvard's digital repository for scholarly articles, theses and dissertatinos, and other Harvard-affiliate generated literature. Harvard Library makes the bibliographic data openly available for all uses, with a standard set of APIs.

### York University Library

🏠 http://bit.ly/1j56kms

The York University Library provides access to a number of library resources important to students. A research guide function allows export of topic-specific and course-specific resource lists targeted to a student's registration selections and program affiliations. Additional functions provide lists of library staff with expertise and topical focus relevant to a student's program and details of current status for the student's library account.

I'm hopeful for libraries, on and off campus, playing a leading role in the world of APIs. The core values that make an API initiative successful reflect the same values that come from the library sciences profession and its evolution over the last couple centuries. When it comes to establishing healthy patterns for API usage, I think we will be looking to libraries more and more in the coming years, as some parts of the tech sector appear have less than honorable intentions when it comes to APIs.

## Other Motivations For Providing Public APIs

There are three other examples of institutions who have deployed APIs, ranging from providing access to student union membership, ePayment, and student information systems. The primary reason that these types of implementations exist is usually because an existing IT system provides some sort of an API that comes bundles with the application.

🏠 http://bit.ly/1soxDda

### University of British Columbia

UBC ePayment is an enterprise level service offered by UBC IT that allows participating UBC departments/merchants to process financial transactions online. Depending on the needs of the department, merchants can sign up for one or more of the three key ePayment components to process payments: Web Service, Student Financial Account and Virtual Terminal. Merchants are able to search/view these transactions, issue back refunds, and view payment reports via the ePayment Adminstrative System. All ePayment transactions are automatically deposited or withdrawn from departmental speedchart or account as ePayment is integrated with UBC Financial Management System (FMS).

### University of Toronto

The EASI API (Enterprise Applications and Solutions Integration API) was developed to allow staff members of the University of Toronto to programatically access the ROSI (Repository of Student Information). ROSI publishes via the web, using protocols that are XML-based. Unlike other web services designed for machine to machine interaction on a network, EASI is developed only as the service is needed. Web services are accessed through a SOAP interface.

🏠 http://bit.ly/1ugVD1g

### University of Warwick

The University of Warwick student union membership API allows you to secure your own website by validating that members of your club, society or other organsiation are who they say they are. Each member that wants to validate themselves on your website must then generate their own key that is unique to their membership of the organisation. The tool to do this will appear on your page as soon as your organisation key has been created.

🏠 http://bit.ly/1j56hqO

I would confidently say that most universities already have existing services running, and with minimal effort tey could be made available in a way that is consistent with modern web API management practices. There are numerous ways to educate IT staff that existing systems have APIs, and the importance of making them available to not just internal IT operations, but also to faculty, students, and outside vendors. I’ll talk later about ways to encourage access to existing APIs either through top down leadership, or in a more organic way, from the students and faculty.

## Central Developer Program

A single place to find all of an institutions API and developer resources, usually at developer.example.edu, or within a specific folder at example.edu/developer, is essential to a heightened viability of any campus API effort. A single portal for secure, machine readable access to all of an institutions digital assets, provides a consistent place for anyone to find what they need, anytime.

Employing a central registry for all digital services is a good idea when viewed from a campus IT perspective. Also investing in this registry not being just a technical portal, where only seasoned IT professionals and developers go, but also any average user, is essential to injecting consistency, and cultivating an essential human tone to any IT operations. In the last decade, various mini-evolutions in technology like Web 2.0, Cloud Computing, and APIs have slowly pulled IT out of the shadows, and into the public space, and campus IT needs to get on board with the latest API evolution or risk falling behind.

Faculty and administrators should all know they can visit the centralized campus developer area and find not just open data, and API resources, but also tools and services they can put to use in their daily work. An API developer area is not just for programmers, there are many buttons, badges, widgets, connectors, spreadsheet integrators, and other tools that allow for non-developers to put APis to work for them—any modern campus developer program should provide a robust buffet of services any faculty member can take advantage of, with very little guidance, in a self-service way.

An increasing number of college students are coming to school, equipped with an awareness that APIs exist, from their exposure to popular technology platforms like Twitter, Facebook, and Instagram. All of these leading technology providers have centralized developer areas, and often provide direct integration with the Software as a Service (SaaS) accounts of student-users. This type of pollination between software users, and modern web API programs have taught users that they can find their own buttons, badges, widgets and put other API driven services like If This Then That, and Zapier to use when managing their digital identity.

Higher education institutions will need to look beyond the classic definition of what a web service or API is, and realize that they provide increasing value to not just internal IT, but also the faculty, administrators, and students across an institution.  Providing a single, common place to find digital assets is quickly becoming the norm, not just in leading tech companies, but within federal, state and city governments, nonprofit organizations, and eventually across higher education institutions.

## Providing The Resources API Consumers Depend On

The current web API movement has been building for almost 14 years, started by API pioneers like SalesForce and Amazon, resulting in some pretty proven models for what types of resources API consumers need for a successful integration. When you look at the central developer areas for SalesForce and Amazon, but also for new players like Twilio and SendGrid, you see some consistent resources present between all of them, ensuring API consumers have what they need for building web, mobile, and single page applications, and increasingly to connect to physical devices in our every day world.

The minimum viable resources you need to support a central API developer program are:

- **Overview** - A simple, clear, and concise description of what a developer program is intended to do, with introduction to the API resources that are available within a developer program.
- **Getting Started -** Easy to follow, step by step instructions on how to get started using an API, and to take an application from idea to production with as little friction as possible.
- **Self Service Registration -** Even if developers need further approval to get full access, some sort of self-service, 24/7 access to a developer area is the standard protocol for most API platforms--requiring developers to register, but can get at least limited access to APIs at any time.
- **API Reference / Documentation -** Intuitive, simple, and up to date documentation of all API endpoints, and the authentication process is standard procedure, with many APIs providing also delivering some sort of interactive documentation using formats like Swagger or API Blueprint.
- **Code Samples & SDKs -** An suite of code samples, libraries, and SDKs should be made available in top programming languages like PHP, Python, Ruby, JavaScript, Java, C#, and possibly leading languages like Go, for developers to put to use in their integrations.

- **Blog -** An active blog, with RSS, provides a single communication channel that API consumers can depend on to receive updates about API and its parent companies operation.
- **Forum -** Forums, either self-hosted by the API provider, or taking advantage of existing public forums like Stack Exchange, provide an indirect channel for developers to obtain support from the API provider, or from their community of peers.
- **Access Levels & Pricing -** Providing a transparent summary of API access levels, alongside any pricing that applies to the levels of API access available, provides clear information for API consumers to consider as they plan their integrations.
- **Terms and Conditions -** Covering the legal aspects of API operations in a way that protects the API providers interests, as well as allowing API consumers enough room to build and operate businesses around their API integrations, is a standard practice within all API developer programs.

There are other bells and whistles you can add in, like a Twitter account or putting Github to work, but API pioneers like SalesForce, Flickr, Twitter and Google have established a common set of essential API building blocks through trial and error over the last decade—leaving this simple checklists that thousands of other API providers follow, making it a default for all APIs.

Internal, student, or external vendors will look to a centralized developer program, containing a consistent number of resources, delivered in a way that matches what people have grown accustom to, when integrating with APIs across the public and private sector over the last decade.

## Migration From A SOA To An API Way Of Life

The modern web API movement was born out of the enterprise service oriented architecture methodology, also known as SOA. This earlier, and still very relevant methodology, is a way to deploy web services, in a very technical, top down, IT governance kind of way. Around 2000 what we know as the modern web API began to take a different direction with a series of API deployments from like SalesForce, Amazon and EBay--something that was further evolved by social emdia platforms like Twitter, Facebook, and tech giants like Google, then again by Amazon, resulting what is now known as cloud computing.

As I mentioned before, most institutions use web services as part of regular systems and IT operations, but very few actually understand the benefits of breaking free from this very silo-ed, and technical way of operating, and see the benefits a web API approach. It is very likely that most IT campus operations have a registry of web services, existing software systems that have APIs which are not being put to use, and most importantly lack a centralized, organized vision for making services available outside of internal IT operations.

It takes a forward thinking IT group, that is aware of what has been happening across the tech landscape over the last 14 years, with APIs. This type of transition in thinking can be seen playing out at Brigham Young University with their over 250+ APIs, which once you scratch the surface you realize that the majority are older SOAP services, part of a legacy SOA strategy, with only a handful of them being redesigned to reflect modern web APIs. Even with this awareness, BYU is still just beginning to get a handle on what a centralized developer area will look like, let alone an organized and consistent approach to designing, deploying and managing APIs.

The enterprise has begun waking up to the benefits of an API strategy over a purely SOA driven strategy, and I’d say that city, state, and federal government have started doing the same in the last two years, which is something I predict higher education institutions will be doing over the next five years. Evolving from a very internally focused IT vision of web services, to an externally focused API vision, one that includes not just IT, but administrators, faculty, students, and outside vendors and third party developers will not be easy for many schools, but it will be required to stay relevant in coming years.

## Providing Robust Internal IT Access To APIs

Using web services to deliver digital services across campus should be nothing new to seasoned IT personnel. However there will be a learning curve when it comes to the nuances around making the same assets and resources available via web APIs. Traditional IT approaches to deliver web services, and even some web APis are often very technically driven by the view from the IT or developer staff. One of the hallmarks of most leading web APIs, is that they focus delivering digital resources in a way that is closer to what the end-user will actually need, making what is often a very technical resource, much more accessible to a wider audience.

Self-service, secured access for IT staff to all services, across all systems enables a much more flexible, agile, and resilient environment, that eliminates redundancy, and encourages consistency. Modern web APIs use the same infrastructure as websites, so it should not require too much retooling to switch to designing, deploying and managing web APIs from a legacy SOA infrastructure.

In an ideal university API effort, campus IT will take the lead on all aspects of the API program, but IT groups should also be aware, as with other examples of shadow IT, where faculty, administrators and students look to external software services, to meet their needs—the same can happen with your API program, leaving you with less control than if campus IT took the lead.

## Providing Faculty Access To Campus APIs

University faculty and administrators are increasingly depending on technology to do their job. As institutions continue to require staff to use common systems like the Learning Management System (LMS), and the Student Information System (SIS), contact, content, document, media, and a variety of other systems, they need to also understand the importance ensuring that all systems enable feature, setting, content and data portability that can be introduced by using APIs.

APIs could be as simple as allowing a teacher to custom pull a student roster for a course in a different way than the student information system will allow, or using APIs could be part of a larger research project, allowing a distributed group of researchers to work on a single document or even database via APIs. APIs are often right below the surface on many of the systems faculty and administrators are already depending on for their job, and it would not take much to make these existing APIs available in a way that allows staff to use across multiple channels including the web, mobile, single page applications, spreadsheets, as well as across external data analysis and visualization tools.

## Providing Student Access To Campus APIs

The thought of providing direct access to campus digital resources for students will seem like an insane idea to many IT leaders, but in 2014, it is a new generation of delivering information technology and services, and with all entering freshman being born and raised on the Internet, we can know longer exclude them from the flow of campus digital information.

Many students are aware that APIs exist in the platforms they use like Twitter and Instagram, and posses a unique digital literacy that many campus IT staff, and teachers will not have. The energy of each wave of students that pass through an institution needs to be captured, and making campus resources and services available through simple, secure APIs can go a long way to provide the access students will need to find workarounds to frustrating IT systems, and even innovate when building applications and tools that schools do not have the time or resources to do themselves.

Providing access to campus resources for students is a new idea, but one that makes sense in the Internet age, where schools are being required to do more with less, making it unacceptable to not capture the energy of the young, technically savvy students making their way through our universities. in coming years, making APIs available to students will not just be necessary to streamline campus operations, it will be an essential part of preparing students for the workplace they are about to enter, and assert their role in the digital economy, in a way that makes them aware and confident of their digital presence.

## Putting Public APIs to Use On Campus

APIs within the university are not just about providing access to campus resources, it can be about consuming public APIs like Flickr, Twitter, Tumblr and WordPress in the classroom, or during administrative operations. APIs are designed to be part of the open web, allowing for the development of web, mobile, and single page applications that can span both campus, and public API resources in a single blog post, mobile application, or internal Student Information System(SIS).

The line between internal and external services has blurred with the growth of the Internet, and the websites, web applications, and mobile applications that are used on campus reflect this reality—there is no reason APIs will be any different. Every outside cloud service that is used on campus should have an accompanying API, allowing for integration with existing systems, as well as feature and data portability for students, faculty, administrators and IT.

It is likely that campus IT has already started organizing outside software and services into a central place for faculty, administrators and students to find, providing a sanctioned list of outside services that can be put to use. IT needs to do the same with the APIs that these services offer--especially if they are not campus owned. APIs are everywhere, whether or not an institution acknowledges them, and it is better to not just take a pro-active stance and make them part of regular activity, or better yet, maybe take a lead and make sure students are not just prepared, but fully equipped for their role in our increasingly digital world.

## Explore Using APIs In The Classroom

Whether its external, publicly available APIs like Twitter and Flickr, or internal resources like the campus events calendar or student directory, there are opportunities for incorporating APIs into the classroom. Blogs, equipped with RSS have been accepted for some time on campus, and APIs are just the next logical evolution from RSS for aggregating not just content, but data, and other valuable programmatic resources or audio, and media services.

I recently worked with the University of Arlington, in Texas (UTA), and edX, the online education partnership between Harvard and MIT, and other institutions like UTA, on how APIs can be applied in an online data & analytic course. In this scenario, students could be introduced to valuable data resources like US census data, corporate records using OpenCorporates, creative commons pictures from Flickr, or maybe just Twitter data--in a single, online class experience.

There are some interesting opportunities to incorporate both public, and campus API resources into the classroom experience, adding an entirely new dimension to the university API discussion. The skills students will learn by using APIs in the classroom can help increase overall web literacy, and their proficiency in finding what they need to get their homework done, and some day give them an advantage in the workplace as well.

## Investing In Hackathon, Data and Process Jams

Universities are following the lead of the tech sectors and employing hackathons to provide a rich, competitive environment for students to learn about pubic, and campus APIs, build websites, web and mobile applications that help them in their classes, support a project, or solve a problem they face in their everyday campus life.

University of California Berkeley is experimenting with integrating hackathons into courses, as well as campus life. Instructors are playing with the best way to ensure that the intellectual exhaust from each hackathon is captured, and any ideas, code, and other projects have the opportunity to live on as part of the classroom or campus experience. Imagine a world where students help build, and improve on the applications they use to interact with their schools. UC Berkeley has acquired applications from students, setting an interesting new precedent for how campus IT services are delivered, and moving us closer to this vision.

Hackathons are also being employed in an intercollegiate way, allowing schools to compete with each other when it comes to developing applications, much like sports, and other campus activities. Hackathons are a simple format, that is attractive to young students, and can be done very cost effectively, a regular basis, using existing classroom space, and campus catering services.

Hackathons are also being employed beyond the startup community, and being executed at the industry level in education, healthcare, and energy, as well as within government from city to the federal government. Hacking events provide a great networking opportunity for students, an excellent way to introduce them to new technology, while also potentially developing code, applications, and new process that can improve campus operations, and enrich the education experience

## External Vendor Access To Services

Beyond internal IT, faculty and administrators, and even the students, APIs can securely open up access to valuable campus resources, in a way that allows campus IT to monitor, and shut off access to prevent abuse. While information about university APIs can be public, the APIs are most often secured using “API keys”, or employs oAuth when personally identifiable information (PII) is being accessed, allowing for students and faculty to have a say in who has acces to their personal information.

Modern web API management infrastructure allows for the securing, and metering of web APis, which requires any external vendor to first register for access, obtain API keys, before they can access any campus resources. This type of access allows campus IT to easily open up access to outside vendors, while protecting the privacy and security of campus operations—striking a balance when it comes to building website, web and mobile applications around campus resources.

Take a look at UC Berkeley’s developer area to get an idea of what a centralized developer area, with a modern API management platform looks like. UC Berkeley now has a single location to hang any campus API, as well as a single point of entry for internal and external access to campus API resources, with different levels of access if you are student, faculty, or an external vendor being given access to API to build an application, or accomplish a specific systems integration.

## Implementing A Top Down API Strategy

After reviewing the approach of the six universities who are deploying centralized API strategies, there is one approach that stands out as what I’d call a top down approach, which you can see playing out in the BYU API program--where you have an effort, being led by the CIO office, with buy-in from executive level leaders.

This top down approach at BYU is definitely an ideal situation for deploying APIs at a university because you have direct access to all IT resources, and can go big, like they have at BYU, making almost every part of campus operations accessible through the over 250+ APIs they have available.

Even with this level of access, there is still a lot of work to do, bringing together the BYU developer area, delivering the supporting resources that API consumer will need, and maintaining top level buy in, but overall it is a pretty impressive start, compared to other institutions—providing a pretty good example of why a top down API strategy for universities can be the way to go.

## Resorting To A Bottom Up API Strategy

In contrast to the top down approach at BYU, you can see a more bottom up, organic approach playing out at UC Berkeley. While there is buy-in for a centralized developer area, and API management platform, the original effort has been led by a small group of faculty, administrators, and students. After building momentum and aggregating a handful of valuable campus APIs, the program has managed to gain momentum, through slowly educating, and including other individuals, programs, and administrators across the UC system.

A bottom up approach at UC Berkeley is definitely not ideal, but in reality, will probably be how most APIs are introduced on many campuses around the world. It isn’t just higher education institutions that are target of this type of bottom up pressure to open up access to digital resources--large companies are feeling industry pressure, and government has been responding to civic pressure for government to be more open and transparent the last couple years.

My dream implementation at a university would be both top down, and bottom up. The grassroots evangelism that occurs bottom up is really necessary to get the buy in across a large institution. In our federal government the president mandated that agencies open up their data and resources, but it is taking a grassroots army of evangelists, data stewards, and passionate developers to actually make it a reality on the ground—this same story will play out across universities over the next couple of years.

## Higher Education Is Front Line For Web Literacy

A large part of our worlds have moved online. Whether it is entertainment at home, education in the classroom, or out and about on our mobile devices, we are living more of our lives online—increasing the imperative that we are empowering every student with at least a minimum amount of web literacy. Think of web literacy, like understanding your personal finances, every single adult must have a certain level of awareness of how our financial system operates. You don't need to understand the inner workings of banking and global markets, but you need to know how to setup a bank account, apply for credit or debit cards, balance your checkbook and pay your taxes. As our lives move online and our digital footprint expands, our data is fast becoming the online currency of Internet platforms like Facebook, Twitter and Pinterest, increasing the need for us to understand the mechanisms at play in this new digital economy, and empower each student to take back some control.

Web literacy goes beyond just what you will need to know in your job and career, it is what you will need to be a literate citizen, and use government services, conduct business, and even interact with your friends and community around you. Just as higher education has played a significant role in preparing our youth for the real world in the past, it will have to take the lead in providing them with the basic web literacy skills they will need to engage online in everyday scenarios, as well as the specialty career trajectories. API is a layer of web literacy, one that allows the tech savvy individual to peel back how the Internet works, and configure, tweak, and make the Internet work for them--not the other way around.

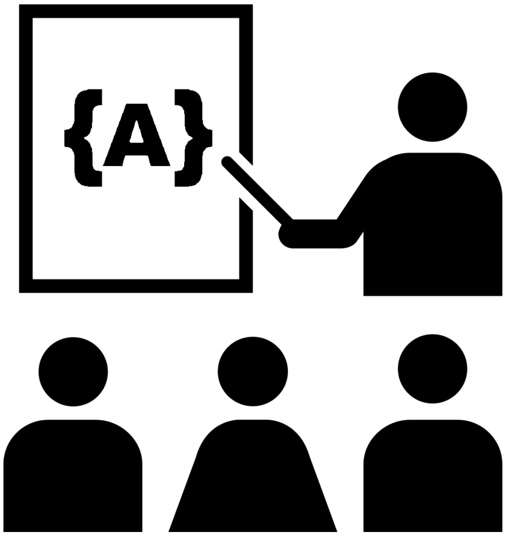
## Providing A Blueprint For Other University To Follow

My goal with this white paper is to create a static snapshot of the current state of APIs in higher education, share some of what I learned looking through the leading universities in the space, and also apply some of my wider knowledge from the API industry beyond the university campus--in hopes university leaders are listenging.

I want this paper to be a blueprint that any university CIO or CTO, student or energetic professor can pick up, and get an idea of what is going on across other universities when it comes to APIs. I'm fascinated by what I'm seeing out of BYU, UW, UC Berkeley, and UM, and as I do with the rest of the space, I'm looking forward to monitoring their progress, and having more conversations about how we can make university API efforts more successful.

I will continue to update this white paper as I find more examples of institutions putting to APIs to work. While much of this research is rooted in what I'm seeing across these 12 higher education institutions, there is still much that is speculation based on my experience, and as I find more concrete examples in the wild I'll expand and add to this paper.

I can't emphasize enough, the important role that higher education will play in not just equipping the next generation with the web literacy skills they will need to be successful, but also help take the lead in defining the underlying pipes that will drive our increasingly digital economy. APIs are not good by default, they can be used is some very bad ways, but I'm confident that if higher education takes a lead in the space, the patterns that can be established will provide a healthier path forward than if we let Silicon Valley continue to exclusively lead.



# Appendix A: Curated News and Resources

- **Should universities care about APIs? | Jisc** from www.jisc.ac.uk on 3/3/2014), full resource available at http://www.jisc.ac.uk/blog/should-universities-care-about-apis-15-jan-2013
- **List of 15+ University APIs** from blog.mashape.com on 12/17/2013), full resource available at http://blog.mashape.com/list-of-15-university-apis/
- **Syracuse University's iSchool IT Girls want to change the world with data** from blogs.splunk.com on 11/19/2013), full resource available at http://blogs.splunk.com/2013/11/19/syracuse-universitys-ischool-it-girls-want-to-change-the-world-with-data/
- **Reshaping Scholarly Communication: UC Open Access Policy** from osc.universityofcalifornia.edu on 8/2/2013), full resource available at http://osc.universityofcalifornia.edu/open-access-policy/
- **Free Webinar: Evolving Strong Authentication at the University of Arizona** from www.duosecurity.com on 6/21/2013), full resource available at https://www.duosecurity.com/blog/free-webinar-evolving-strong-authentication-at-the-university-of-arizona
- **Budget transparency for an open university** from community.openspending.org on 6/16/2013), full resource available at http://community.openspending.org/2013/06/universities-on-the-spending-map/
- **University guide 2014: download the Guardian tables and see how the rankings have changed** from www.theguardian.com on 6/4/2013), full resource available at http://www.theguardian.com/news/datablog/2013/jun/04/university-guide-2014-table-rankings
- **Dogfooding the OrgSync API: How We Built the User Conference Statistics Page** from blog.orgsync.com on 6/2/2013), full resource available at http://blog.orgsync.com/2013/how-we-built-user-conference-statistics-page
- **Association of Commonwealth Universities selects Huddle to manage the Chevening Scholarships programme** from www.huddle.com on 5/15/2013), full resource available at http://www.huddle.com/blog/association-of-commonwealth-universities/
- **DecisionDesk pays out colleges via Dwolla** from blog.dwolla.com on 3/12/2013), full resource available at http://blog.dwolla.com/decisiondesk-pays-out-colleges-via-dwolla/
- **University of Waterloo Open Data API** from api.uwaterloo.ca on 6/5/2012), full resource available at http://api.uwaterloo.ca/