{"logo":"API Evangelist"}

API Industry Guide

API Deployment

August 2015

by Kin Lane, the API Evangelist

This is meant to be a field guide to the fast changing world of API deployment, providing you an overview of companies, tooling, common building blocks, and some of the latest news from across the landscape.

An Overview of API Deployment

Just like there are many types of APIs, there are many approaches to deployment APIs. The why and how of API deployment varies widely, and until recently, has been a conversation that was left to developer and IT groups. With the latest wave of growth in the world of web APIs, we are seeing an expansion of the conversation around how we deploy APIs on-premises, and in the cloud.

The history of API deployment reflects its roots in IT, and amongst web and mobile developers. You either deployed your API using a gateway (IT led), or you hand crafted your own, or more likely used an open source API framework for your scaffolding (developer led). Both approaches have their benefits, but some organizations may not have the resources to afford a proper gateway, or to develop their own custom API -- resulting in a wave of cloud, and open source solutions for rapidly deploying APIs from database, spreadsheets, and much more.

The goal of this research guide is to help businesses be more aware of high level concepts at play when it comes API deployment, before making an investment. I wanted to provide business and organizational leaders with the overall knowledge they need to understand the potentially different paths that are available for API deployment. Whether you choose to bootstrap the initiative, or look for an API management service provider to assist, this project is meant to get you up to speed, with a healthy awareness from across space--at a 100K view.

We will walk you through the challenges of doing it yourself, how you can employ API frameworks, gateways, as well as a variety of cloud based, API deployment solutions. As the space is expanding I will also look at the evolving solutions for deploying APIs from spreadsheets, databases, and search indexes. There is a wealth of services, open tooling, and blueprints for API deployment to show you--almost too many to keep up with.

After reviewing this API deployment project you should be understand the overview of API deployment and ready to think about API management.

The Do It Yourself Approach To API Deployment

When faced with API deployment many developers groups size up the situation and then decide to do it themselves, there is really not much to it, they declare! When in reality there is a lot of nuance, and detail to consider when crafting a consistent, stable API for a production environment--something you are better off not re-inventing. This is one reason why we have the

diverse, bespoke amount of APIs we have in the space currently, that don't work very well together--too many people re-inventing the wheel.

Even if you can't afford a high-end gateway solution, or even the emerging variety of cloud-based solutions, there are too many open source frameworks available when deploying APIs--it just doesn't make sense to do this yourself in 2015.

For now, let's explore everything beyond doing it yourself, and move on to the next best thing, using a common framework.

Developer Led Usage of Frameworks for API Deployment

For many API deployment initiatives, using an existing open source API framework in your desired programming language is the way to go. If a company has the available resources to deploy and maintain websites, it can easily deploy and manage API infrastructure using any of the existing API frameworks.

An API framework can make basic API deployment from a data source, something a developer can tackle in minutes, rather than hours or days. You and your technical talent will have to select the API framework that suits your development style, but most are intuitive and easy to adopt. It is likely that your developers, whether internal, or 3rd party, are already using a framework for deploying web sites, web applications, or mobile applications, that can also being driving your API backend.

I will showcase an even wider variety of open source frameworks below under tooling, in this section I wanted to showcase a handful of frameworks that have a higher profile, and usually a company and /or organized effort behind them.

- **Apache Wink** - Apache Wink is a simple yet solid framework for building RESTful Web services. It is comprised of a Server module and a Client module for developing and consuming RESTful Web services. The Wink Client module is a Java based framework that provides functionality for communicating with RESTful Web services. The framework is built on top of the JDK HttpURLConnection and adds essential features that facilitate the development of such client applications.

    - **Website:** https://wink.apache.org/
    - **Github:** https://github.com/apache/wink/

- **Django REST** - Django REST framework is a powerful and flexible toolkit that makes it easy to build Web APIs. It offers an attractive, web browseable version of your API,

and the option of returning raw JSON. The Django Rest Framework provides powerful model serialization, display data using standard function based views, or get granular with powerful class based views for more complex functionality. All in a fully REST compliant wrapper.

    - **Website:** http://www.django-rest-framework.org

- **OpenRasta** - OpenRasta is an open-source .NET framework for building everything web, from web sites to RESTful APIs. Written from the ground-up with testing in mind, OpenRasta removes the faff from your code and lets you concentrate on writing clean, beautiful code. Proven in the real world, OpenRasta is the perfect choice for building scalable, high-performance, reliable web services.

    - **Website:** http://openrasta.org/
    - **Github:** https://github.com/openrasta

- **Restlet** - The Restlet API Platform enables developers and non-developers to design, create, run and manage the APIs that provide access to any data or application. Restlet Framework is the most widely used open source solution for Java developers who want to create and use APIs. The first Platform-as-a-Service dedicated to web APIs, APISpark enables any organization to become an API provider in minutes via an intuitive browser interface.

    - **Website:** https://apispark.com/
    - **Twitter:** https://twitter.com/apispark
    - **Blog:** http://blog.restlet.com/
    - **Github:** https://github.com/apispark

I will keep migrating other frameworks to this section, from the open source tooling portion below, as I see frameworks evolve, and grow. It can be difficult to assess the usage and scope of any single API framework and its community. My goal is to just give myself a variety of definitions to work within, which is something that pushes me to keep studying, and understanding each company, organization, or project--hopefully it will do the same for you.

Using API Gateways for Deployment

API gateways have been on the market for some time now, a carry-forward from the earlier SOA evolution of APIs. It can be tough to understand the features each gateway offers, and their

abilities to act as mediators between the enterprise resources and their consumers. API gateways are a powerful way to address API deployment, and if this route makes sense for your company, you should reach out to the providers, have a conversation, and stick with the one provider that speak to your company's language.

All of these API gateways have kept with the times, evolving from their SOA roots, but now also understanding the importance of delivering RESTful APIs, and provide lightweight JSON from a variety of existing legacy, internal resources. Additionally there is a handful of these that have been born in this new web API realm, and do not know anything of SOA--resulting in some very new views on what is a gateway.

API Gateways are the API deployment option for larger businesses, ones that have larger volumes of existing resources to expose. Deploying individual APIs with frameworks, and hand-rolling your own management tools can quickly become overwhelming, and these providers have hardened their offerings, making them a solid option. You see cloud providers like Amazon and Google quickly delivering in this area as the opportunity for API deployment becomes more evident.

All of this movement is resulting in a pretty seismic shift in area of API gateways lately, part of which is being led by Amazon with their recent gateway release, but is also being pushed into new areas, with companies like JustAPIs. Just exactly what is an API gateway, where you deploy it, and who can put it to use is rapidly being redefined--making this ongoing research very interesting in my opinion.

This API gateway portion of my monitoring still feels like a catch all bucket for anything that calls itself a gateway. With future iterations of this paper, I will work to further break out, as I get to know each of the solutions better, and discover other nuances. For now, you get this mixed bag of API gateways.

- **3scale** - 3scale APIcast lets you deploy on a cloud API gateway service in just a few clicks, without modifications to backend code. APIcast is free to use and supports up to 50,000 API transactions per day, making it the perfect solution for low- or moderate-volume APIs. Enterprises can use APIcast for staging and quick testing.

    - **Website:** http://www.3scale.net/api-management/apicast/
    - **Twitter:** https://twitter.com/3scale
    - **Blog:** http://www.3scale.net/blog/
    - **Github:** https://github.com/3scale

- **Akana** - Monitor activity for specific partners, apps developers and evaluate their impact on your business. Analyze which aspects of your APIs are being adopted, which devices or apps are being built against your APIs, study reasons for defections and incorporate insights from this analysis to improve your APIs and create best practices for developers to avoid common pitfalls with your APIs.

    - **Website:** http://www.akana.com/
    - **Twitter:** https://twitter.com/akanainc
    - **Blog:** https://blog.akana.com/

- **Amazon Web Services** - Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a front door for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any Web application.

    - **Website:** http://aws.amazon.com
    - **Twitter:** https://twitter.com/awscloud
    - **Blog:** http://aws.amazon.com/blogs/aws/
    - **Github:** https://github.com/amazonwebservices

- **Apache Knox** - The Apache Knox Gateway is a REST API Gateway for interacting with Hadoop clusters. The Knox Gateway provides a single access point for all REST interactions with Hadoop clusters. In this capacity, the Knox Gateway is able to provide valuable functionality to aid in the control, integration, monitoring and automation of critical administrative and analytical needs of the enterprise.

    - **Website:** https://knox.apache.org/

- **ApiAxle** - ApiAxle is a proxy that sits on your network, in front of your API(s) and manages things that you shouldn't have to, like rate limiting, authentication and caching. It's fast, open and easy to configure. ApiAxle is different to the cloud based services such as Mashery in that itu2019s intended to be installed within your LAN and be managed by you. This means you own your users, you own your data and you can more easily manage costs. ApiAxle is open-source. This means you can

modify it as much as you like and contribute changes back. Others will do the same and gradually the system will become all the better for it.

    - **Website:** http://apiaxle.com/
    - **Twitter:** https://twitter.com/apiaxle
    - **Blog:** http://blog.apiaxle.com/
    - **Github:** https://github.com/apiaxle

- **Apigee** - Apigee delivers an intelligent API platform to accelerate the pace of digital business. We help companies - from disruptive start-ups to the Fortune 100 u2013 use their enterprise data and services to create connected digital experiences for customers, partners and employees. This is digital business.

    - **Website:** https://apigee.com
    - **Twitter:** https://twitter.com/apigee
    - **Blog:** https://blog.apigee.com/front
    - **Github:** https://github.com/apigee

- **Axway** - Axway (Euronext: AXW.PA), acquired API Gateway vendor, Vordel, to complement its existing MFT and B2B Gateway products. Axway's suite of products enables enterprise to govern the flow of data within and across the edge of the enterprise, unlocking the tremendous value this can bring to business interactions. Axway API Management offers the enterprise-grade API management architecture with the security to protect sensitive data, control access and support integration to a wide range of on-premise and cloud-based applications.

    - **Website:** http://www.axway.com/vordel-products/
    - **Twitter:** https://twitter.com/axway
    - **Blog:** http://blogs.axway.com/

- **Computer Associates** - The industry-leading family of API gateways from CA Technologies offers unmatched flexibility, performance and security. Building on the foundation of its industry-leading SOA application gateway technology for exposing, securing and managing backend applications, network systems or infrastructure via APIs, CA Technologies has added critical mobile, cloud and REST composition features. With this additional functionality, CA API Gateways represent the best available solution for enterprises looking to open data and services to partners,

developers, mobile apps, cloud services and smart devices.

    - **Website:** http://www.ca.com
    - **Twitter:** https://twitter.com/cainc
    - **Blog:** http://blogs.ca.com/?intcmp=footernav

- **JustAPIs** - JustAPIs is designed to help developers overcome existing limitations when it comes to building APIs. The JustAPIs solution is a high-performance, compiled executable that can run on any Linux, Windows, or Mac OSX based hardware, from a single developer's laptop to large-scale, clustered production environments. With zero dependencies, JustAPIs can be installed instantly and includes sample APIs with familiar JavaScript-based business logic, so developers can be up and running with their own API server in minutes.

    - **Website:** http://justapis.com/
    - **Twitter:** https://twitter.com/justapis

- **NGINX** - With the explosion of APIs within applications, itu2019s critical to ensure they are protected, tracked, and monetized. NGINX Plus is a trusted platform to manage and secure HTTP-based API traffic. Leveraged by leading API management platforms, NGINX Plus allows you to deliver API-based services with speed, reliability, scalability, and security.

    - **Website:** https://www.nginx.com/solutions/api-gateway/
    - **Twitter:** https://twitter.com/nginxorg

- **OpenLegacy** - OpenLegacy has pioneered a disruptive approach to legacy modernization and application integration by focusing on business needs instead of on replacing existing solutions. The company believes in building on the value of legacy systems rather than rushing to replace them. OpenLegacy's comprehensive, end-to-end integration platform unlocks the application services and information in mainframes, AS/400 and databases, automatically transforming them into mobile, web and cloud applications.

    - **Website:** http://openlegacy.com/
    - **Twitter:** https://twitter.com/openlegacy

    - **Blog:** http://openlegacy.com/blog/
    - **Github:** https://github.com/openlegacy

- **StrongLoop** - StrongLoop develops StrongLoop Suite, a leading Mobile API Tier along with being the primary code contributor to Node.js. StrongLoop Suite includes an open source private mBaaS, an operations console and a supported package of Node.js, containing advanced debugging, clustering and support for private npm registries. StrongLoop was founded by Node core-contributors, Enterprise mobile architects and veterans of open source and Could companies.

    - **Website:** https://strongloop.com/node-js/api-gateway/
    - **Twitter:** https://twitter.com/strongloop
    - **Blog:** http://strongloop.com/strongblog/
    - **Github:** https://github.com/strongloop

- **TIBCO Software** - TIBCO API Exchange Gateway governs third-party access to enterprise SOA systems by federating heterogeneous services and providing a single point of control. It lets you extend your enterprise applications and services onto cloud, partner ecosystem, and mobile platforms by managing and enforcing policies such as security, throttling, transformation, routing, and monitoring.

    - **Website:** http://www.tibco.com/
    - **Twitter:** https://twitter.com/tibco
    - **Blog:** http://www.tibco.com/blog/

- **Tyk** - An open source, lightweight, fast and scalable API Gateway. Set rate limiting, request throttling, and auto-renewing request quotas to manage how your users access your API. Tyk supports access tokens, HMAC request signing, basic authentication and OAuth 2.0 to integrate old and new services easily. Tyk can record and store detailed analytics which can be segmented by user, error, endpoint and client ID across multiple APIs and versions. Integrate your existing or new applications with Tyk using a simple REST API, Tyk even support hot-reloads so you can introduce new services without downtime.

    - **Website:** http://tyk.io/
    - **Blog:** http://tyk.io/blog/

- **WaveMaker, Inc.** - WaveMaker, Inc. provides WaveMaker Enterprise u2013 An aPaaS software for rapid application delivery of enterprise custom apps. WaveMaker Enterprise provides benefits such as visual rapid application development, out of the box support for security, web services integration and data modelling, high performance cloud platform based on Docker containers, a self service management console that provides simplified administration and many more such features.

    - **Website:** http://www.wavemaker.com/
    - **Twitter:** https://twitter.com/wavemaker
    - **Blog:** http://www.wavemaker.com/blog/
    - **Github:** https://github.com/wavemaker

To be honest, API gateways never excited me much. I feel like they do not provide API developers with the tools they truly need to get to know an underlying resource, and empower them to craft high quality APIs. However with the latest round of API gateway additions, my opinions are changing. For the first time in over 10 years, the API gateway conversation is thawing, and potentially moving forward in a way that benefits small API providers, all the way up to the enterprise.

The Emerging Cloud API Deployment Platforms

Cloud computing has been a reality for some years now. While APIs were essential to the development of cloud computing itself, it is also a natural course for companies to also offer API deployment services that run exclusively in the cloud.

A handful of companies have emerged in the last couple years to provide API Deployment as a Service, connecting existing and new data sources, then generating web APIs, often complete with common API management services to boot.

Cloud API deployment from common data sources will make API deployment accessible to the masses, making the API deployment of numerous resources easier for developers, while also making API deployment something a non-developer could potentially handle.

- **Amazon Web Services** - Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a front door for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any Web

application.

    - **Website:** http://aws.amazon.com
    - **Twitter:** https://twitter.com/awscloud
    - **Blog:** http://aws.amazon.com/blogs/aws/
    - **Github:** https://github.com/amazonwebservices

- **Apigility** - Separating presentation logic from data provides the flexibility to support multiple client form factors, and future-proofs apps to allow behind-the-scenes change without breaking user interfaces. With Apigility, you can take the code that powers your business, and then API-enable it.

    - **Website:** https://apigility.org/
    - **Twitter:** https://twitter.com/apigility
    - **Github:** https://github.com/zfcampus

- **Apitite** - Apitite is a web application programming interface (API) creation and management service on the cloud. Apitite enables businesses to securely expose their existing data to their users, employees, and partners automatically. Through our web interface, clients can create web APIs by providing connection details to their datastore, specifying which data to expose, and listing who should be granted access to the data. In addition, clients can access usage analytics to observe how their users are consuming their data. Apitite takes care of the web hosting, security, and user access of each API.

    - **Website:** https://www.apitite.net/
    - **Twitter:** https://twitter.com/apititeapi
    - **Blog:** http://blog.apitite.net/

- **BinaryOps.io** - Separating presentation logic from data provides the flexibility to support multiple client form factors, and future-proofs apps to allow behind-the-scenes change without breaking user interfaces. With Apigility, you can take the code that powers your business, and then API-enable it.

    - **Website:** http://www.binaryops.io/
    - **Twitter:** https://twitter.com/binaryops
    - **Blog:** http://www.binaryops.io/blog/

- **Deployd** - Deployd is an open source API design, and deployment platform that allows developers to quickly design, customize, and deploy an API, with supporting application interface via a downloadable app, and command line utilities. Deployd is a downloadable solution, you can use on your desktop.

    - **Website:** http://deployd.com/
    - **Twitter:** https://twitter.com/deploydapp
    - **Github:** https://github.com/deployd

- **DreamFactory** - DreamFactory Software is a private, venture-backed company based in Campbell, California, with an additional development center in Atlanta, Georgia. We were founded in 2005 and funded in 2006 by New Enterprise Associates. Today, DreamFactory publishes the DreamFactory Services Platform, an open source REST API platform for mobile application developers.

    - **Website:** http://www.dreamfactory.com/
    - **Twitter:** https://twitter.com/dfsoftwareinc
    - **Blog:** http://blog.dreamfactory.com/
    - **Github:** https://github.com/dreamfactorysoftware

- **Espresso Logic** - Espresso Logic provides the fastest way to create RESTful APIs to multiple backend data sources u2013 with both read and write capabilities. You create an enterprise-class API complete with logic using reactive programming and fine-grain security. And API creation is declarative. Deliver backends u2013 not just APIs u2013 in 1/10 the time and cost.

    - **Website:** http://www.espressologic.com/
    - **Twitter:** https://twitter.com/sqlrest
    - **Blog:** http://www.espressologic.com/blog/
    - **Github:** https://github.com/espressologiccafe

- **Google** - Create RESTful services and make them accessible to iOS, Android and Javascript clients. Automatically generate client libraries to make wiring up the frontend easy. Built-in features include denial-of-service protection, OAuth 2.0 support and client key management.

    - **Website:** https://cloud.google.com/endpoints/

    - **Github:** https://github.com/google

- **Instant API** - Instant API is Platform as a Service that makes it easy for Developers to Create, Host, Publish & Document APIs from Existing Data and Services. What used to take weeks or months now takes minutes and is a fraction of the cost.

    - **Website:** http://instantapi.co/
    - **Twitter:** https://twitter.com/instantapi
- **Restlet** - The Restlet API Platform enables developers and non-developers to design, create, run and manage the APIs that provide access to any data or application. Restlet Framework is the most widely used open source solution for Java developers who want to create and use APIs. The first Platform-as-a-Service dedicated to web APIs, APISpark enables any organization to become an API provider in minutes via an intuitive browser interface.

    - **Website:** https://apispark.com/
    - **Twitter:** https://twitter.com/apispark
    - **Blog:** http://blog.restlet.com/
    - **Github:** https://github.com/apispark

- **Service Stack** - ServiceStack started development in 2008 with the mission of creating a best-practices services framework with an emphasis on simplicity and speed, reducing the effort in creating and maintaining resilient message-based SOA Services and rich web apps.

    - **Website:** https://servicestack.net
    - **Twitter:** https://twitter.com/servicestack
    - **Github:** https://github.com/servicestack

- **StackHut** - StackHut turns code into live APIs, powered by containers. Take any code - eg. Python, JS, native - describe your config, and run stackhut deploy. Your functions and classes are automatically turned into APIs, accessible through a POST request. Powered by Linux containers, it can now scale to meet any demand.

    - **Website:** https://stackhut.com
    - **Twitter:** https://twitter.com/stackhut
    - **Github:** https://github.com/stackhut

- **StrongLoop** - StrongLoop develops StrongLoop Suite, a leading Mobile API Tier along with being the primary code contributor to Node.js. StrongLoop Suite includes an open source private mBaaS, an operations console and a supported package of Node.js, containing advanced debugging, clustering and support for private npm registries. StrongLoop was founded by Node core-contributors, Enterprise mobile architects and veterans of open source and Could companies.

    - **Website:** http://strongloop.com/
    - **Twitter:** https://twitter.com/strongloop
    - **Blog:** http://strongloop.com/strongblog/
    - **Github:** https://github.com/strongloop

- **SwiftIQ** - SwiftIQ provides web-service application programming interface (API) infrastructure to facilitate data accessibility and predictive analytics through the Swift Access and Swift Predictions products. Swift Access is an award-winning backend platform to unify and secure disconnected data then deliver and analyze it on-demand to power real-time digital actions. Swift Predictions allows users to apply adaptive, machine learning algorithms to discover insights fast and make applications smarter.

    - **Website:** http://www.swiftiq.com/
    - **Twitter:** https://twitter.com/swiftiq
    - **Blog:** http://www.swiftiq.com/blog
    - **Github:** https://github.com/swiftiq

- **WaveMaker, Inc.** - WaveMaker, Inc. provides WaveMaker Enterprise u2013 An aPaaS software for rapid application delivery of enterprise custom apps. WaveMaker Enterprise provides benefits such as visual rapid application development, out of the box support for security, web services integration and data modelling, high performance cloud platform based on Docker containers, a self service management console that provides simplified administration and many more such features.

    - **Website:** http://www.wavemaker.com/
    - **Twitter:** https://twitter.com/wavemaker
    - **Blog:** http://www.wavemaker.com/blog/
    - **Github:** https://github.com/wavemaker

- **Zatar** - Zatar is a product of Zebra Technologies, a global leader respected for innovation and reliability. Zebra provides products and services that enable real-time visibility into organization's assets. Zebra solutions support Enterprise Asset Intelligence and the Zatar enterprise IoT platform is a perfect example! Zatar provides a standards-based approach to connectivity and control of devices along with open APIs to create apps, onboard devices and enable collaboration.

    - **Website:** http://www.zatar.com/
    - **Twitter:** https://twitter.com/zatar_iot
    - **Blog:** http://www.zatar.com/about
    - **Github:** https://github.com/zatar-iot

I am still working to better define exactly what these companies do. Cloud API deployment is a pretty big bucket, so look for these to move around pretty quickly. I feel like many of them will also keep pivoting until they find the right approach that solves problems for API providers, while also bringing in revenue.

API Deployment with Focus on the Database

There are many reasons for deploying an API, but the most common reason I'd say is that you want to provide access to information stored in a database. While many API deployment providers allow for you to connect to a database, some providers have a stronger focus in this area, and I want to highlight them separately.

With the popularity of Hadoop, and emergence of Lambda from AWS, I'd say we will see the definition of what is “database”, and what is API deployment from these databases, rapidly evolve. For now, here is what I'm tracking on in the area of API deployment using databases.

- **Apitite** - Apitite is a web application programming interface (API) creation and management service on the cloud. Apitite enables businesses to securely expose their existing data to their users, employees, and partners automatically. Through our web interface, clients can create web APIs by providing connection details to their datastore, specifying which data to expose, and listing who should be granted access to the data. In addition, clients can access usage analytics to observe how their users are consuming their data. Apitite takes care of the web hosting, security, and user access of each API.

    - **Website:** https://www.apitite.net/
    - **Twitter:** https://twitter.com/apititeapi

    - **Blog:** http://blog.apitite.net/

- **CKAN** - CKAN is a powerful data management system that makes data accessible u2013 by providing tools to streamline publishing, sharing, finding and using data. CKAN is aimed at data publishers (national and regional governments, companies and organizations) wanting to make their data open and available.

    - **Website:** http://ckan.org/
    - **Blog:** http://ckan.org/blog/

- **Espresso Logic** - Espresso Logic provides the fastest way to create RESTful APIs to multiple backend data sources u2013 with both read and write capabilities. You create an enterprise-class API complete with logic using reactive programming and fine-grain security. And API creation is declarative. Deliver backends u2013 not just APIs u2013 in 1/10 the time and cost.

    - **Website:** http://www.espressologic.com/
    - **Twitter:** https://twitter.com/sqlrest
    - **Blog:** http://www.espressologic.com/blog/
    - **Github:** https://github.com/espressologiccafe

- **Fluidinfo** - Fluidinfo are the makers of FluidDB, an online cloud database. Using a flexible underlying representation of information and a new model of control, FluidDB allows users and applications to work with information more naturally. That includes dynamically organizing, sharing, combining and augmenting information, and searching in ways that have previously not been possible. It also allows users to choose exactly which information to share with whom, with separate controls for reading and writing. FluidDB lets data be social.

    - **Website:** http://fluidinfo.com/
    - **Twitter:** https://twitter.com/fluidinfo
    - **Blog:** http://blogs.fluidinfo.com/
    - **Github:** https://github.com/fluidinfo

- **Orchestrate** - Orchestrate provides a simple RESTful API service that eliminates the need for developers to operate their own databases in production. Powered by best-of-breed NoSQL databases, it unifies full-text search, social graph, activity feed, key/value document and time-ordered event queries in a single interface, eliminating

the operational and financial burden of deploying databases when both building new applications and adding new features to existing ones.

    - **Website:** http://orchestrate.io/
    - **Twitter:** https://twitter.com/orchestrateio
    - **Blog:** http://orchestrate.io/blog/
    - **Github:** https://github.com/orchestrate-io

- **SlashDB** - SlashDB connects your internal databases and constructs a REST/HTTP web service, easily making database content accessible by URLs for getting, updating, inserting and deleting in a secure way. SlashDB provides connectors for Microsoft SQL Server, Oracle, MySQL, PostGreSQL, IBM DB2 and Sybase--covering the top 5 databases you will find in the enterprise or small to medium businesses. SlashDB automatically turns databases into online resource so their content becomes accessible to authorized web, mobile and enterprise applications for reading and writing under standard data formats. Technically speaking, it makes REST APIs out of relational databases.

    - **Website:** http://www.slashdb.com/
    - **Twitter:** http://twitter.com/slash_db

API deployment empowers API providers to go beyond just resource based API design, and allow them to evolve their designs, and deploy much more meaningful endpoints, going beyond just the table and column names of its source. APIs provide us a way to abstract away the complexity of query language, and the raw resources beneath, and craft simple, high value endpoints that can be applied in web, mobile, and other type of application or system integration with ease.

API Deployment Using Spreadsheets

Most of the world's business data is stored and manipulated within spreadsheets, so it makes sense that we would seek to deploy APIs from spreadsheets. I've seen a number of these types of API services providers come and go, but there is a current wave of them that are making a go of it.

These are the companies I'm tracking on that make API deployment from spreadsheets something that anyone can do -- no developer skills required.

- **CKAN** - CKAN is a powerful data management system that makes data accessible u2013 by providing tools to streamline publishing, sharing, finding and using data. CKAN is aimed at data publishers (national and regional governments, companies and organizations) wanting to make their data open and available.

    - **Website:** http://ckan.org/
    - **Blog:** http://ckan.org/blog/

- **Restlet** - APISpark is an cloud API platform that lets you create, host, manage and use web APIs. Using the Restlet Framework at its core, APISpark simplifies the web API experience, the time to market, and the overall cost to get started and to scale you APIs. Restlet is a web API platform vendor, pioneer of RESTful web APIs. APISpark serves our customers around the world, providing software to build web APIs, which includes APISpark, the PaaS version of Restlet.
    - **Website:** https://apispark.com/
    - **Twitter:** https://twitter.com/apispark
    - **Blog:** http://blog.restlet.com/
    - **Github:** https://github.com/apispark

- **Sheetlabs** - Sheetlabs turn spreadsheets into well-documented APIs, quickly and easily. Many businesses still share product lists, service coverage areas, or other tabular information with one another using spreadsheets attached to email. There are so many things that can go wrong with this. Sheetlabs was built with the goal of making it really easy for businesses to build APIs on top of their spreadsheets.

    - **Website:** https://sheetlabs.com
    - **Twitter:** https://twitter.com/sheetlabs

- **Socrata** - Socrata is the leading developer and provider of Open Data Services, a category of cloud-based Web 2.0 solutions that enable federal, state, and local governments to dramatically improve the reach, usability and social utility of their public information assets. The Socrata Social Data Platform is a turnkey information delivery platform that reduces lifecycle management costs for government customers while boosting their ability to disseminate relevant information and data-driven services to a wide range of audiences including citizens, civic application developers, researchers, journalists and internal stakeholders.

    - **Website:** http://socrata.com

    - **Twitter:** https://twitter.com/socrata
    - **Blog:** http://www.socrata.com/tech-blog/
    - **Github:** https://github.com/socrata

In theory you could launch an API by proxying the Google Sheets API, and the Office 365 API, but the services above make it much easier, and something a non-developer could potentially do. As we see the awareness of APIs grow amongst the average business user, we will see API deployment solutions from spreadsheets become more commonplace.

APIs Generated From Search Indexes

Another source of quick, consistent, and potentially highly valuable APIs, is through indexing document and other data sources using search indexing solutions like Lucene. While these are not cloud solutions, they are potentially something a non-developer could do with a number of SaaS providers emerging that support Lucene and Elastic.

- **Apache Solr** - Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest internet sites.

    - **Website:** http://lucene.apache.org/solr/

- **Elastic** - Elasticsearch is on a mission to organize data and make it easily accessible. The company delivers the world’s the most advanced open source search and analytics engine available and make real-time data exploration available to anyone. By having a laser focus on achieving the best user experience imaginable, Elasticsearch has become one of the most popular and rapidly growing open source solutions in the market. Today, Elasticsearch is used by thousands of enterprises in virtually every industry. We take good care of our customers and users, providing production support, development support and training worldwide.

    - **Website:** http://www.elasticsearch.org/
    - **Twitter:** https://twitter.com/elasticsearch
    - **Blog:** http://www.elasticsearch.org/blog/
    - **Github:** https://github.com/elasticsearch

I have seen some pretty impressive API deployments in government, where they took a high value data file, or series of data files, and indexed with Elastic, and deployed a pretty sophisticated API using the built in tooling. I'd like to spend time diving into more of these types of implementations and try to identify some common patterns, but that is often the beauty of APIs--you just can't see what is behind.

Functional API Deployment Using Code

The most common type of API is all about consuming, and serving up data. However there are many functional APIs that apply code and algorithms to make the magic happen. I'm seeing a new trend emerge, where API deployment service providers are allowing you to upload, paste, or connect to code that lives on Github, and they wrap it, and launch an endpoint that puts the algorithm to work as an API.

This is a new area I am carving off, so I am just getting started, but here are some of the solutions for quickly deploying APIs from code.

- **Algorithmia** - We're building a community around state-of-the-art algorithm development. Users can create, share, and build on other algorithms and then instantly make them available as a web service.

    - **Website:** https://algorithmia.com
    - **Twitter:** https://twitter.com/algorithmia
    - **Blog:** http://blog.algorithmia.com/

- **Apitite** - Apitite is a web application programming interface (API) creation and management service on the cloud. Apitite enables businesses to securely expose their existing data to their users, employees, and partners automatically. Through our web interface, clients can create web APIs by providing connection details to their datastore, specifying which data to expose, and listing who should be granted access to the data. In addition, clients can access usage analytics to observe how their users are consuming their data. Apitite takes care of the web hosting, security, and user access of each API.

    - **Website:** https://www.apitite.net/
    - **Twitter:** https://twitter.com/apititeapi
    - **Blog:** http://blog.apitite.net/

- **StackHut** - StackHut turns code into live APIs, powered by containers. Take any code - eg. Python, JS, native - describe your config, and run StackHut deploy. Your functions and classes are automatically turned into APIs, accessible through a POST request. Powered by Linux containers, it can now scale to meet any demand.

    - **Website:** https://stackhut.com
    - **Twitter:** https://twitter.com/stackhut
    - **Github:** https://github.com/stackhut

If this area of API deployment can merge with the whole containerization world, we will really start seeing the server-side reality of the last 15 years melt away. A new world of API deployment solutions that allow you to paste any code language you want, and launch a standard web API interface on demand.

Web Scraping As A Data Source For API

Harvesting or scraping of content and data from other public web sources is something many do, but few talk publicly about. While scraping does infringe on copyright in some situations, in many others situations, it is quickly becoming a legitimate way to acquire content or data, for use as an API.

There is a lot of content available online, where the current stewards do not have the control, resources, or interest in making content available in a machine readable format. In these scenarios, for many developers, if you want the content, you just write a scrape script, and liberate it from the site, to be used as you wish.

In response to this, we are seeing a new breed of API service providers emerge, who assist users in deploying APIs from data and content that is liberated through harvesting or scraping. For the first time, I'm seeing enough of these new tools and services, that I have broken them out into its own area, and make part of this API deployment guide.

- **Import.io** - Import.io turns the web into a database, releasing the vast potential of data trapped in websites. Allowing you to identify a website, select the data and treat it as a table in your database. In effect transform the data into a row and column format. You can then add more websites to your data set, the same as adding more rows and query in real-time to access the data.

    - **Website:** http://docs.import.io/
    - **Twitter:** https://twitter.com/importio

    - **Blog:** http://blog.import.io/
    - **Github:** https://github.com/import-io

- **Kimono Labs** - Kimono is a way to turn websites into structured APIs from your browser in seconds. You don't need to write any code or install any software to extract data with Kimono. The easiest way to use Kimono is to add our bookmarklet to your browser's bookmark bar. Then go to the website you want to get data from and click the bookmarklet. Select the data you want and Kimono does the rest.

    - **Website:** https://www.kimonolabs.com/
    - **Twitter:** https://twitter.com/kimonolabs
    - **Blog:** http://blog.kimonolabs.com/

I have a whole other research area dedicated to scaping, which has other API driven solutions for harvesting content from sites. The service providers listed here are specifically focused on deploying APIs from content that is scraped from websites.

While scraping of websites to some may seem an undesirable, or even illegal way to gather your data and content, but there are plenty of legitimate use cases where it is the only option.

Everyone Is Doing Microservices

While doing this research, we cannot forget about one of the hottest conversations going on in the world of APIs right now -- microservices. Whether you subscribe to the ideology or not, there is still a lot to learn from the movement when it pertains to API deployment.

Along with the wave of conversations, I'm seeing a new breed of API deployment providers emerge, speaking the microservice way. These are the ones I've added so far.

- **Dockpit** - Painless isolation for your (micro)service development process. Dockpit makes it trivial to develop your (micro)service in isolation. It mocks the APIs you depend on and puts data stores, message queues and service registries in predictable states.

    - **Website:** https://dockpit.io/
    - **Twitter:** https://twitter.com/dockpit
    - **Github:** https://github.com/dockpit

- **Seneca** - A Micro-Services toolkit for Node.js. This toolkit lets you write clean code that you can scale without needing to refactor. Start with everything in one process, and split it all out onto multiple systems when you need to.

    - **Website:** http://senecajs.org/
    - **Twitter:** https://twitter.com/senecajs
    - **Github:** https://github.com/rjrodger/seneca

Microservices is another area that I am just getting going on, and seeing rapidly evolve. We'll see if microservice focused API deployment providers continue to pick up momentum this year, If I do I'll keeping updating future editions with anything I come across

A Wealth of Open Source Tooling

There is a huge amount of open tooling out there for helping you deploy your APIs. Too much for me to keep up with, but I do try to add any framework, gateway, and other open source software I come across.. Early on the only thing that was on the list was open source frameworks in a variety of languages, but now I have quite a list of tools, broken down into different groups, as I work to make sense of it all.

API Blueprint

- **Dockpit API Blueprint Template** (https://github.com/dockpit/pit-api-blueprint/)- A Dockpit template for mocking Rest HTTP APIs using API Blueprint

Cold Fusion

- **Taffy** (http://atuttle.github.com/taffy/)- Taffy is a ColdFusion framework that helps you build RESTful web services with very little boilerplate code, very little configuration, and to be honest, very little effort.

Django

- **Django Rest** (http://django-rest-framework.org/)- Django REST framework is a lightweight REST framework for Django, that aims to make it easy to build well-connected, self-describing RESTful Web APIs.nnnnAutomatically provides an awesome Django admin style browse-able self-documenting API.nClean, simple, views for Resources, using Django's new class based views.nSupport for ModelResources with out-of-the-box default implementations and input

validation.nPluggable parsers, renderers, authentication and permissions - Easy to customise.nContent type negotiation using HTTP Accept headers.nOptional support for forms as input validation.nModular architecture - MixIn classes can be used without requiring the Resource or ModelResource classes.nnn

Framework

- **Bottle** (http://bottlepy.org/docs/dev/)- Bottle is a fast, simple and lightweight WSGI micro web-framework for Python. It is distributed as a single file module and has no dependencies other than the Python Standard Library.
- **CherryPy** (http://cherrypy.org/)- CherryPy is a pythonic, object-oriented HTTP framework, that allows developers to build web applications in much the same way they would build any other object-oriented Python program. This results in smaller source code developed in less time.nCherryPy is now more than six years old and it is has proven very fast and stable. It is being used in production by many sites, from the simplest ones to the most demanding ones.
- **Dave** (https://github.com/evantahler/php-dave-api)- DAVE is a minimalist, multi-node, transactional API framework written in PHP, which contains an end-to-end API test suite for TDD, a Task model, an Active Database Model, and a stand-alone development server (PHP) to get you started.DAVE is an acronym that stands for Delete, Add, Edit, and View. These 4 methods make up the core functionality of many transactional web applications.The DAVE API aims to simplify and abstract may of the common tasks that these types of APIs require. DAVE does the work for you, and hes not CRUD. Dave was built to be both easy to use, but to be as simple as possible.Dave contains an end-to-end API test suite for TDD, a Task model, an Active Database Model, and a stand-alone development server (written in just PHP) to get you started.
- **Django Rest** (http://django-rest-framework.org/)- Django REST framework is a lightweight REST framework for Django, that aims to make it easy to build well-connected, self-describing RESTful Web APIs.nnnnAutomatically provides an awesome Django admin style browse-able self-documenting API.nClean, simple, views for Resources, using Django's new class based views.nSupport for ModelResources with out-of-the-box default implementations and input validation.nPluggable parsers, renderers, authentication and permissions - Easy to customise.nContent type negotiation using HTTP Accept headers.nOptional support

for forms as input validation.nModular architecture - MixIn classes can be used without requiring the Resource or ModelResource classes.nnn

- **Epiphany** (https://github.com/jmathai/epiphany#readme)- The Epiphany framework is fast, easy, clean and RESTful. The framework does not do a lot of magic under the hood. It is, by design, very simple and very powerful.The documentation provides a few conventions that we believe lead to well written code but youre free to use any style youd like. The framework never dictates how you should write or structure your application.
- **Falcon** (http://falconframework.org/)- Falcon is a ridiculously fast, minimalist Python web framework for building cloud APIs and app backends. The Falcon web framework encourages the REST architectural style, meaning (among other things) that you think in terms of resources and state transitions, which map to HTTP verbs.
- **Flask** (http://flask.pocoo.org/docs/)- Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: Its BSD licensed!nFlask's documentation is divided into different parts. They recommend that you get started with Installation and then head over to the Quickstart. Besides the quickstart there is also a more detailed Tutorial that shows how to create a complete (albeit small) application with Flask. If you'd rather dive into the internals of Flask, check out the APIdocumentation. Common patterns are described in the Patterns for Flask section.nFlask depends on two external libraries: the Jinja2 template engine and the Werkzeug WSGI toolkit. These libraries are not documented here.
- **Flight** (http://flightphp.com/)- Flight is a fast, simple, extensible framework for PHP. Flight enables you to quickly and easily build RESTful web applications.
- **FRAPI** (http://getfrapi.com/)- FRAPI is a new type of framework that embraces the standards and mindset of how the web is modeled. Instead of being a general purpose framework like the Zend Framework, Symfony, Cake, Lithium, etc. which are great for building web applications, FRAPI aims at removing the whole frontend layer complexity that handling REST calls can bring.nFRAPI is a RESTful API Framework that allows developers to rapidly develop RESTful APIs that are easily scalable and highly performant. FRAPI consists of two specific parts: The administration interface and the public API.
- **Grape** (http://rdoc.info/github/intridea/grape)- Grape is a REST-like API micro-framework for Ruby. Its designed to run on Rack or complement existing web application frameworks such as Rails and Sinatra by providing a simple DSL to easily develop RESTful APIs. It has built-in support for common conventions,

including multiple formats, subdomain/prefix restriction, content negotiation, versioning and much more.

- **Limonade** (http://www.limonade-php.net/)- Limonade is a PHP micro framework for rapid web development and prototyping. It's inspired by frameworks like Sinatra or Camping in Ruby, or Orbit in Lua. It aims to be simple, lightweight and extremly flexible.
- **Pylons** (http://www.pylonsproject.org/)- The Pylons Project was founded by the people behind the Pylons web framework to develop web application framework technology in Python. Rather than focusing on a single web framework, the Pylons Project will develop a collection of related technologies. The first package is the Pyramid web framework.
- **Recess** (http://www.recessframework.org/)- Recess is a RESTful PHP framework that can be used by both beginner and seasoned developers. Recess is fast, light-weight, and has a very small footprint—ideal for LAMP development and drag-and-drop deployment to shared hosts. Recess is a modern framework that uses a loosely-coupled Model-View-Controller architecture designed and optimized specifically for PHP 5.
- **rest!** (http://engineering.silk.co/post/90354057868/announcing-rest-a-haskell-rest-framework)- rest is a set of packages used to write, document, and use RESTful applications. You write your API in Haskell using rests DSL. This API can then be run in different web frameworks like happstack, snap, or wai. Additionally, you can automatically generate documentation from it, as well as client libraries for Haskell and Javascript.
- **Resteasy** (http://www.jboss.org/resteasy)- Resteasy is a JBoss.org project aimed at providing productivity frameworks for developing client and server RESTful applications and services in Java. It is mainly a JAX-RS implementation but youll find some other experimental code in the repository.
- **RestFixture** (https://github.com/smartrics/restfixture/wiki)- The RestFixture is a FitNesse fixture that allows developers, testers, and/or product owners to write test fixtures for REST API with simplicity in mind. The idea is to write tests that are self-documenting and easy to write and read, without the need to write Java code. The fixture allows test writers to express tests as actions (using any of the allowed HTTP methods) to operate on resource URIs and to express expectations about the content of the return code, headers and body. All without writing one single line of Java code!

- **Restify** (http://mcavage.github.com/node-restify/)- restify is a node.js module built specifically to enable you to build correct REST web services. It intentionally borrows heavily from express as that is more or less the de facto API for writing web applications on top of node.js.
- **RESTkit** (http://restkit.org/)- RestKit is an Objective-C framework for iOS that aims to make interacting with RESTful web services simple, fast and fun. It combines a clean, simple HTTP request/response API with a powerful object mapping system that reduces the amount of code you need to write to get stuff done.
- **Restler** (http://luracast.com/products/restler/)- A RESTful API server framework that is written in PHP that aids your mobile / web / desktop applications. A framework, but with a difference – Restler is all here to bend and mend to your needs.nWriting Server made easy and light. With the light weight, Restler makes writing a server as easy as writing it with just 3 PHP files.nRestler's advantage is the simplicity. You can create a PHP class with some functions to expose. If you know how to write object oriented PHP, then you already know how to use Restler. It's action speaks for its effectiveness.nRestler is all about being light and easy. All public methods are automatically mapped to a URL Tailor-made Restler is known for the customization options. You just need to write class and add methods in PHP. It is just there!
- **Restlet** (http://www.restlet.org/) - Restlet is a lightweight, comprehensive, open source REST framework for the Java platform. Restlet is suitable for both server and client Web applications. It supports major Internet transport, data format, and service description standards like HTTP and HTTPS, SMTP, XML, JSON, Atom, and WADL. A GWT port of the client-side library is also available.
- **RESTRack** (http://restrack.me/)- A Model-View-Controller Framework. RESTRack follows the MVC design pattern that you are already familiar with. It is inspired by Rails and follows a few of its conventions. But, while Rails is a powerful tool for full web applications, RESTRack is targeted at super lightweight data services.nRack aims to provide a minimal API for connecting web servers and web frameworks. RESTRack leverages Rack to provide a minimal framework to create REST services. From the get go, RESTRack was designed to make it extremely easy to develop performant REST data services.nRESTRack is perfect for data generation. Rich JavaScript frameworks such as ExtJS, jQuery UI, dojo, and native mobile applications would be well suited to for RESTRack serving as the data layer. The framework has a very small memory footprint, making it a great choice for cloud type architectures.

- **RESTx** (http://restx.mulesoft.org/)- RESTx is a light-weight open-source platform for the creation of RESTful data access and integration resources and web services. It emphasizes simplicity, sane defaults and out-of-the-box usability. No complex configuration, no steep learning curve: You will be up and running in just 5 minutes.nRESTx is not your usual application framework and can simplify the creation of RESTful web services and resources.
- **Roar** (https://github.com/apotonick/roar)- Roar is a framework for parsing and rendering REST documents. Nothing more. With Roar, REST documents – also known as representations – are defined using a new concept called representers. Both syntax and semantics are declared in Ruby modules that can be mixed into your domain models, following clean OOP patterns.nRoar comes with built-in JSON, JSON::HAL and XML support. It exposes a highly modular architecture and makes it very simple to add new media types and functionality where needed. Additional features include client HTTP support, coercion, client-side caching, awesome hypermedia support and more. Representers fit pretty well in DCI environments, too.nRoar is completely framework-agnostic and loves being used in web kits like Rails, Webmachine, Sinatra, Padrino, etc. Actually, Roar makes it fun designing real, hypermedia-driven, and resource-oriented systems that will even make Steve sleep happily at night so he finally gets some REST!
- **Seam REST** (http://seamframework.org/seam3/restmodule)- Seam REST is a lightweight module that aims to provide additional integration with technologies within the Java EE platform as well as third party technologies. Seam REST is independent of CDI and JAX-RS implementations and thus fully portable between Java EE 6 environments.
- **Slim** (http://slimframework.com/)- What began as a weekend project became a simple yet powerful PHP 5 framework to create RESTful web applications. The Slim micro framework is everything you need and nothing you don't. Slim lets you build a complete PHP web service with only a single PHP file. Features include: RESTful routing, Named routes, Route passing, Route redirects, Route halting, Custom views, HTTP caching, Signed cookies, Custom 404 page, Custom 500 page, Error handling and Logging.
- **Spring Framework** (http://www.springsource.org/spring-framework)- The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the plumbing of enterprise applications so that teams can focus on

application-level business logic, without unnecessary ties to specific deployment environments.

- **Taffy** (http://atuttle.github.com/taffy/)- Taffy is a ColdFusion framework that helps you build RESTful web services with very little boilerplate code, very little configuration, and to be honest, very little effort.
- **Tonic** (http://peej.github.com/tonic/)- Tonic is an open source less is more, RESTful Web application development PHP library, where everything useful is a resource, not a file, not a CGI script, a resource, an abstract concept of something useful that the client wants to grab hold of. Resources are located by URLs, URLs are cheap and form the universal addressing system of the Web. Tonic helps you develop Web applications that embrace the way the Web really works, enabling your applications to scale, extend and work with other systems easily.
- **Wave Framework** (http://www.waveframework.com/)- Wave is a PHP micro-framework that is built loosely on model-view-control architecture and factory method design pattern. It is made for web services, websites and info-systems and is built around a native API architecture, caching and smart image and resource management. Wave is a compact framework that does not include optional libraries, has a very small footprint and is developed keeping lightweight speed and optimizations in mind.
- **web py** (http://webpy.org/)- web.py is a web framework for Python that is as simple as it is powerful. web.py is in the public domain; you can use it for whatever purpose with absolutely no restrictions.
- **Zend** (http://framework.zend.com/manual/en/zend.rest.server.html)- Zend_Rest_Server is intended as a fully-featured REST server. To call a Zend_Rest_Server service, you must supply a GET and POST methods, with a value that is the method you wish to call. You can then follow that up with any number of arguments using either the name of the argument or using arg following by the numeric position of the argument. When returning values, you can return a custom status, you may return an array with each status.

Haskell

- **rest!** (http://engineering.silk.co/post/90354057868/announcing-rest-a-haskell-rest-framework)- rest is a set of packages used to write, document, and use RESTful applications. You write your API in Haskell using rests DSL. This API can then be

run in different web frameworks like happstack, snap, or wai. Additionally, you can automatically generate documentation from it, as well as client libraries for Haskell and Javascript.

Java

- **Resteasy** (http://www.jboss.org/resteasy)- Resteasy is a JBoss.org project aimed at providing productivity frameworks for developing client and server RESTful applications and services in Java. It is mainly a JAX-RS implementation but youll find some other experimental code in the repository.
- **RestFixture** (https://github.com/smartrics/restfixture/wiki)- The RestFixture is a FitNesse fixture that allows developers, testers, and/or product owners to write test fixtures for REST API with simplicity in mind. The idea is to write tests that are self-documenting and easy to write and read, without the need to write Java code. The fixture allows test writers to express tests as actions (using any of the allowed HTTP methods) to operate on resource URIs and to express expectations about the content of the return code, headers and body. All without writing one single line of Java code!
- **Restlet** (http://www.restlet.org/)- Restlet is a lightweight, comprehensive, open source REST framework for the Java platform. Restlet is suitable for both server and client Web applications. It supports major Internet transport, data format, and service description standards like HTTP and HTTPS, SMTP, XML, JSON, Atom, and WADL. A GWT port of the client-side library is also available.
- **RESTx** (http://restx.mulesoft.org/)- RESTx is a light-weight open-source platform for the creation of RESTful data access and integration resources and web services. It emphasizes simplicity, sane defaults and out-of-the-box usability. No complex configuration, no steep learning curve: You will be up and running in just 5 minutes.nRESTx is not your usual application framework and can simplify the creation of RESTful web services and resources.
- **Seam REST** (http://seamframework.org/seam3/restmodule)- Seam REST is a lightweight module that aims to provide additional integration with technologies within the Java EE platform as well as third party technologies. Seam REST is independent of CDI and JAX-RS implementations and thus fully portable between Java EE 6 environments.
- **Spring Framework** (http://www.springsource.org/spring-framework)- The Spring Framework provides a comprehensive programming and configuration model for

modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the plumbing of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

Limonade

- **Limonade** (http://www.limonade-php.net/)- Limonade is a PHP micro framework for rapid web development and prototyping. It's inspired by frameworks like Sinatra or Camping in Ruby, or Orbit in Lua. It aims to be simple, lightweight and extremly flexible.

Microservices

- **Dockpit API Blueprint Template** (https://github.com/dockpit/pit-api-blueprint/)- A Dockpit template for mocking Rest HTTP APIs using API Blueprint

Mock

- **Dockpit API Blueprint Template** (https://github.com/dockpit/pit-api-blueprint/)- A Dockpit template for mocking Rest HTTP APIs using API Blueprint

Node.js

- **Restify** (http://mcavage.github.com/node-restify/)- restify is a node.js module built specifically to enable you to build correct REST web services. It intentionally borrows heavily from express as that is more or less the de facto API for writing web applications on top of node.js.

Objective-C

- **RESTkit** (http://restkit.org/)- RestKit is an Objective-C framework for iOS that aims to make interacting with RESTful web services simple, fast and fun. It combines a clean, simple HTTP request/response API with a powerful object mapping system that reduces the amount of code you need to write to get stuff done.

PHP

- **Dave** (https://github.com/evantahler/php-dave-api)- DAVE is a minimalist, multi-node, transactional API framework written in PHP, which contains an end-to-end API test suite for TDD, a Task model, an Active Database Model, and a stand-alone development server (PHP) to get you started.DAVE is an acronym that stands for Delete, Add, Edit, and View. These 4 methods make up the core functionality of many transactional web applications.The DAVE API aims to simplify and abstract may of the common tasks that these types of APIs require. DAVE does the work for you, and hes not CRUD. Dave was built to be both easy to use, but to be as simple as possible.Dave contains an end-to-end API test suite for TDD, a Task model, an Active Database Model, and a stand-alone development server (written in just PHP) to get you started.
- **Epiphany** (https://github.com/jmathai/epiphany#readme)- The Epiphany framework is fast, easy, clean and RESTful. The framework does not do a lot of magic under the hood. It is, by design, very simple and very powerful.The documentation provides a few conventions that we believe lead to well written code but youre free to use any style youd like. The framework never dictates how you should write or structure your application.
- **Flight** (http://flightphp.com/)- Flight is a fast, simple, extensible framework for PHP. Flight enables you to quickly and easily build RESTful web applications.
- **FRAPI** (http://getfrapi.com/)- FRAPI is a new type of framework that embraces the standards and mindset of how the web is modeled. Instead of being a general purpose framework like the Zend Framework, Symfony, Cake, Lithium, etc. which are great for building web applications, FRAPI aims at removing the whole frontend layer complexity that handling REST calls can bring.nFRAPI is a RESTful API Framework that allows developers to rapidly develop RESTful APIs that are easily scalable and highly performant. FRAPI consists of two specific parts: The administration interface and the public API.
- **Limonade** (http://www.limonade-php.net/)- Limonade is a PHP micro framework for rapid web development and prototyping. It's inspired by frameworks like Sinatra or Camping in Ruby, or Orbit in Lua. It aims to be simple, lightweight and extremly flexible.
- **Recess** (http://www.recessframework.org/)- Recess is a RESTful PHP framework that can be used by both beginner and seasoned developers. Recess is fast, light-weight, and has a very small footprint—ideal for LAMP development and drag-and-drop deployment to shared hosts. Recess is a modern framework that uses

a loosely-coupled Model-View-Controller architecture designed and optimized specifically for PHP 5.

- **Restler** (http://luracast.com/products/restler/)- A RESTful API server framework that is written in PHP that aids your mobile / web / desktop applications. A framework, but with a difference – Restler is all here to bend and mend to your needs.nWriting Server made easy and light. With the light weight, Restler makes writing a server as easy as writing it with just 3 PHP files.nRestler's advantage is the simplicity. You can create a PHP class with some functions to expose. If you know how to write object oriented PHP, then you already know how to use Restler. It's action speaks for its effectiveness.nRestler is all about being light and easy. All public methods are automatically mapped to a URL Tailor-made Restler is known for the customization options. You just need to write class and add methods in PHP. It is just there!
- **Slim** (http://slimframework.com/)- What began as a weekend project became a simple yet powerful PHP 5 framework to create RESTful web applications. The Slim micro framework is everything you need and nothing you don't. Slim lets you build a complete PHP web service with only a single PHP file. Features include: RESTful routing, Named routes, Route passing, Route redirects, Route halting, Custom views, HTTP caching, Signed cookies, Custom 404 page, Custom 500 page, Error handling and Logging.
- **Tonic** (http://peej.github.com/tonic/)- Tonic is an open source less is more, RESTful Web application development PHP library, where everything useful is a resource, not a file, not a CGI script, a resource, an abstract concept of something useful that the client wants to grab hold of. Resources are located by URLs, URLs are cheap and form the universal addressing system of the Web. Tonic helps you develop Web applications that embrace the way the Web really works, enabling your applications to scale, extend and work with other systems easily.
- **Wave Framework** (http://www.waveframework.com/)- Wave is a PHP micro-framework that is built loosely on model-view-control architecture and factory method design pattern. It is made for web services, websites and info-systems and is built around a native API architecture, caching and smart image and resource management. Wave is a compact framework that does not include optional libraries, has a very small footprint and is developed keeping lightweight speed and optimizations in mind.
- **Zend** (http://framework.zend.com/manual/en/zend.rest.server.html)- Zend_Rest_Server is intended as a fully-featured REST server. To call a Zend_Rest_Server service, you must supply a GET and POST methods, with a

value that is the method you wish to call. You can then follow that up with any number of arguments using either the name of the argument or using arg following by the numeric position of the argument. When returning values, you can return a custom status, you may return an array with each status.

Python

- **Bottle** (http://bottlepy.org/docs/dev/)- Bottle is a fast, simple and lightweight WSGI micro web-framework for Python. It is distributed as a single file module and has no dependencies other than the Python Standard Library.
- **CherryPy** (http://cherrypy.org/)- CherryPy is a pythonic, object-oriented HTTP framework, that allows developers to build web applications in much the same way they would build any other object-oriented Python program. This results in smaller source code developed in less time.nCherryPy is now more than six years old and it is has proven very fast and stable. It is being used in production by many sites, from the simplest ones to the most demanding ones.
- **Django Rest** (http://django-rest-framework.org/)- Django REST framework is a lightweight REST framework for Django, that aims to make it easy to build well-connected, self-describing RESTful Web APIs.nnnnAutomatically provides an awesome Django admin style browse-able self-documenting API.nClean, simple, views for Resources, using Django's new class based views.nSupport for ModelResources with out-of-the-box default implementations and input validation.nPluggable parsers, renderers, authentication and permissions - Easy to customise.nContent type negotiation using HTTP Accept headers.nOptional support for forms as input validation.nModular architecture - MixIn classes can be used without requiring the Resource or ModelResource classes.nnn
- **Falcon** (http://falconframework.org/)- Falcon is a ridiculously fast, minimalist Python web framework for building cloud APIs and app backends. The Falcon web framework encourages the REST architectural style, meaning (among other things) that you think in terms of resources and state transitions, which map to HTTP verbs.
- **Flask** (http://flask.pocoo.org/docs/)- Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: Its BSD licensed!nFlask's documentation is divided into different parts. They recommend that you get started with Installation and then head over to the Quickstart. Besides the quickstart there is also a more detailed Tutorial that shows how to create a complete (albeit small) application with Flask. If you'd rather dive into the internals of

Flask, check out the APIdocumentation. Common patterns are described in the Patterns for Flask section.nFlask depends on two external libraries: the Jinja2 template engine and the Werkzeug WSGI toolkit. These libraries are not documented here.

- **Pylons** (http://www.pylonsproject.org/)- The Pylons Project was founded by the people behind the Pylons web framework to develop web application framework technology in Python. Rather than focusing on a single web framework, the Pylons Project will develop a collection of related technologies. The first package is the Pyramid web framework.
- **RESTx** (http://restx.mulesoft.org/)- RESTx is a light-weight open-source platform for the creation of RESTful data access and integration resources and web services. It emphasizes simplicity, sane defaults and out-of-the-box usability. No complex configuration, no steep learning curve: You will be up and running in just 5 minutes.nRESTx is not your usual application framework and can simplify the creation of RESTful web services and resources.
- **web py** (http://webpy.org/)- web.py is a web framework for Python that is as simple as it is powerful. web.py is in the public domain; you can use it for whatever purpose with absolutely no restrictions.

Ruby

- **Grape** (http://rdoc.info/github/intridea/grape)- Grape is a REST-like API micro-framework for Ruby. Its designed to run on Rack or complement existing web application frameworks such as Rails and Sinatra by providing a simple DSL to easily develop RESTful APIs. It has built-in support for common conventions, including multiple formats, subdomain/prefix restriction, content negotiation, versioning and much more.
- **RESTRack** (http://restrack.me/)- A Model-View-Controller Framework. RESTRack follows the MVC design pattern that you are already familiar with. It is inspired by Rails and follows a few of its conventions. But, while Rails is a powerful tool for full web applications, RESTRack is targeted at super lightweight data services.nRack aims to provide a minimal API for connecting web servers and web frameworks. RESTRack leverages Rack to provide a minimal framework to create REST services. From the get go, RESTRack was designed to make it extremely easy to develop performant REST data services.nRESTRack is perfect for data generation. Rich JavaScript frameworks such as ExtJS, jQuery UI, dojo, and native mobile

applications would be well suited to for RESTRack serving as the data layer. The framework has a very small memory footprint, making it a great choice for cloud type architectures.

- **Roar** (https://github.com/apotonick/roar)- Roar is a framework for parsing and rendering REST documents. Nothing more. With Roar, REST documents – also known as representations – are defined using a new concept called representers. Both syntax and semantics are declared in Ruby modules that can be mixed into your domain models, following clean OOP patterns.nRoar comes with built-in JSON, JSON::HAL and XML support. It exposes a highly modular architecture and makes it very simple to add new media types and functionality where needed. Additional features include client HTTP support, coercion, client-side caching, awesome hypermedia support and more. Representers fit pretty well in DCI environments, too.nRoar is completely framework-agnostic and loves being used in web kits like Rails, Webmachine, Sinatra, Padrino, etc. Actually, Roar makes it fun designing real, hypermedia-driven, and resource-oriented systems that will even make Steve sleep happily at night so he finally gets some REST!

The number of open source API frameworks has steadily grown over the last couple of years, while open source also takes root in other areas like design, management, and integration. As the API gateway world expands, I will be taking another look at the open tooling available, and refresh this list with anything relevant.

Common Building Blocks of API Deployment

There are a handful of common building blocks involved with API deployment. Depending on the resources you desire to open up access to, different building blocks will be needed.

These building blocks have been gathered from evaluating many public API providers, tools that have been developed by experienced API developers, and the services being offered and evolved by API service providers.

These building blocks are meant to be generic, modular concepts that will help on-board business or even technical folks to the expanding world of API deployment--something that historically was something IT or developer staff was aware of, but now business leaders from all disciplines need to be aware of.

- **Framework** - There is no reason to hand-craft an API from scratch these days. There are numerous frameworks out their that are designed for rapidly deploying

web APIs. Deploying APIs using a framework is only an option when you have the necessary technical and developer talent to be able to understand the setup of environment and follow the design patterns of each framework. When it comes to planning the deployment of an API using a framework, it is best to select one of the common frameworks written in the preferred language of the available developer and IT resources. Frameworks can be used to deploy data APIs from CSVs and databases, content from documents or custom code resources that allow access to more complex objects.

- **Proxy** - API proxy are common place for taking an existing API interface, running it through an intermediary which allows for translations, transformations and other added services on top of API. An API proxy does not deploy an API, but can take existing resources like SOAP, XML-RPC and transform into more common RESTful APIs with JSON formats. Proxies provide other functions such as service composition, rate limiting, filtering and securing of API endpoints. API gateways are the preferred approach for the enterprise, and the companies that provide services support larger API deployments.
- **Connector** - Contrary to an API proxy, there are API solutions that are proxyless, while just allowing an API to connect or plugin to the advanced API resources. While proxies work in many situations, allowing APIs to be mediated and transformed into required interfaces, API connectors may be preferred in situations where data should not be routed through proxy machines. API connector solutions only connect to existing API implementations are easily integrated with existing API frameworks as well as web servers like Nginx.
- **Hosting** - Hosting is all about where you are going to park your API. Usual deployments are on-premise within your company or data center, in a public cloud like Amazon Web Services or a hybrid of the two. Most of the existing service providers in the space support all types of hosting, but some companies, who have the required technical talent host their own API platforms. With HTTP being the transport in which modern web APIs put to use, sharing the same infrastructure as web sites, hosting APIs does not take any additional skills or resources, if you already have a web site or application hosting environment.
- **Versioning** - There are many different approaches to managing different version of web APIs. When embarking on API deployment you will have to make a decision about how each endpoint will be versioned and maintained. Each API service provider offers versioning solutions, but generally it is handled within the API URI or passed as an HTTP header. Versioning is an inevitable part of the API life-cycle and

is better to be integrated by design as opposed to waiting until you are forced to make a evolution in your API interface.

- **CSV to API** - Text files that contain comma separated values or CSVs, is one of the quickest ways to convert existing data to an API. Each row of a CSV can be imported and converted to a record in a database, and easily generate a RESTful interface that represents the data stored in the CSV. CSV to API can be very messy depending on the quality of the data in the CSV, but can be a quick way to breathe new life into old catalogs of data lying around on servers or even desktops. The easiest way to deal with CSV is to import directly into database, than generate API from database, but the process can be done at time of API creation.
- **Database to API** - Database to API is definitely the quickest way to generate an API. If you have valuable data, generally in 2013, it will reside in a Microsoft, MySQL, PostgreSQL or other common database platform. Connecting to a database and generate a CRUD, or create, read, update and delete API on an existing data make sense for a lot of reason. This is the quickest way to open up product catalogs, public directories, blogs, calendars or any other commonly stored data. APIs are rapidly replace database connections, when bundled with common API management techniques, APIs can allow for much more versatile and secure access that can be made public and shared outside the firewall.
- **Gateway** - API gateways are enterprise quality solutions that are designed to expose API resources. Gateways are meant to provide a complete solution for exposing internal systems and connecting with external platforms. API gateways are often used to proxy and mediate existing API deployments, but may also provide solutions for connecting to other internal systems like databases, FTP, messaging and other common resources. Many public APIs are exposed using frameworks, most enterprise APIs are deployed via API gateways--supporting much larger deployments.
- **Scraping** - Harvesting or scraping of data from an existing website, content or data source. While we all would like content and data sources to be machine readable, sometimes you have just get your hands dirty and scrape it. While I don't support scraping of content in all scenarios, and business sectors, but in the right situations scraping can provide a perfectly acceptable content or data source for deploying an API.
- **Container** - The new virtualization movement, lead by Docket, and support by Amazon, Google, Red Hat, Microsoft, and many more, is providing new ways to package up APIs, and deploy as small, modular, virtualized containers.

- **Github** - Github provides a simple, but powerful way to support API deployment, allowing for publishing of a developer portal, documentation, code libraries, TOS, and all your supporting API business building blocks, that are necessary for API effort. At a minimum Github should be used to manage public code libraries, and engage with API consumers using Github's social features.

Taking You From API Deployment to API Management

I have broken out API design and deployment into individual areas, separate from API management. However they are all intertwined and overlap is inevitable, but it is important to understand the overview of API deployment, from framework to gateway, and where it dovetails with API design and management.

API deployment was born out of SOA and the enterprise, and the API gateways have the longest history in API deployment. But over the last 10 years a wealth of frameworks and scrappier approaches to deploy APIs build on HTTP has evolved considerably.

After 10 years of evolving API frameworks, and RESTful approaches a new breed of API providers have emerged to provide the next generation of API deployments that are driving mobile apps as well as the Internet of Things, while also making API deployment something anyone can do--not just IT, and devs.

The API Evangelist network is focused on educating the masses around the potential of APIs, helping everyone understand that API deployment is something that any savvy business person should have in their wheelhouse, even if you won't be getting your hands dirty with the actual nuts and bolts of it..

Maybe you have a wealth of data locked up in databases that you need to make available for mobile or tablet apps, and you just don't have the internal resources to design, develop and deploy you APIs. Five years ago, APIs were definitely a purely technical initiative, born out of IT and your developer groups. In 2015 anyone with a little tech curiosity can understand what is possible, and begin to put a strategy in place to achieve your goals.

API Deployment News from the Last Couple of Months

If you aren't familiar with what I do as the API Evangelist. I have spent the last five years monitoring the API space, curating links from across the web, and putting them into different buckets. This curation then goes into my research, analysis, and ultimate this guide you are reading.

I've shared just the last 50 links, spanning the 2nd quarter of 2015. A complete list can be found on Github at deployment.apievangelist.com. While most of the links I provide are from 3rd party locations, some of them provide links to my own analysis in the area.

- **Django-tastypie checking of csrf token in requests** (08-14-2015 on python.6.x6.nabble.com) - http://python.6.x6.nabble.com/django-tastypie-checking-of-csrf-token-in-requests-td5169360.html
- **The Continuing Evolution of The API Gateway ·** (08-12-2015 on apievangelist.com) - http://apievangelist.com/2015/08/11/the-continuing-evolution-of-the-api-gateway/
- **The Continuing Evolution of The API Gateway ·** (08-11-2015 on apievangelist.com) - http://apievangelist.com/2015/08/11/the-continuing-evolution-of-the-api-gateway/
- **JustAPIs - Build REST APIs for Mobile, Web & IoT Apps** (07-17-2015 on justapis.com) - http://justapis.com/
- **Amazon Release API Gateway, a Managed Service to Build and Run APIs** (07-12-2015 on www.infoq.com) - http://www.infoq.com/news/2015/07/aws-api-gateway
- **Build an API under 30 lines of code with Python and Flask** (07-12-2015 on impythonist.wordpress.com) -https://impythonist.wordpress.com/2015/07/12/build-an-api-under-30-lines-of-code-with-python-and-flask/
- **Developing an Apache Wink REST Client for Bluemix Liberty Runtime** (07-12-2015 on developer.ibm.com) -https://developer.ibm.com/bluemix/2015/07/10/developing-apache-wink-rest-client-bluemix-liberty-runtime/
- **Azure/autorest · GitHub** (07-11-2015 on github.com) - https://github.com/azure/autorest

- **A first glimpse of the new Amazon API Gateway** (07-11-2015 on api-university.com) - http://api-university.com/blog/api-architecture/a-first-glimpse-of-the-new-amazon-api-gateway/
- **A first glimpse of the new Amazon API Gateway** (07-11-2015 on api-university.com) - http://api-university.com/blog/a-first-glimpse-of-the-new-amazon-api-gateway/
- **switzersc/api-in-a-box · GitHub** (07-10-2015 on github.com) - https://github.com/switzersc/api-in-a-box
- **Amazon Web Services API Gateway: Why it could be a big deal** (07-10-2015 on www.zdnet.com) - http://www.zdnet.com/article/amazon-web-services-api-gateway-why-it-could-be-a-big-deal/
- **New API Platform Targets Specialized Web and Mobile App Development** (07-10-2015 on www.programmableweb.com) -http://www.programmableweb.com/news/new-api-platform-targets-specialized-web-and-mobile-app-development/brief/2015/07/10
- **The New AWS API (Gateway): Anyone Who Does Not Do This, Will Be Fired. Thank You. Have A Nice Day! - Jeff Bezos ·** (07-09-2015 on apievangelist.com) - http://apievangelist.com/2015/07/09/the-new-aws-api-gateway-anyone-who-does-not-do-this-will-be-fired-thank-you-have-a-nice-day--jeff-bezos
- **Amazon API Gateway – Build and Run Scalable Application Backends** (07-09-2015 on aws.amazon.com) - https://aws.amazon.com/blogs/aws/amazon-api-gateway-build-and-run-scalable-application-backends/
- **Amazon API Gateway – Build and Run Scalable Application Backends** (07-09-2015 on aws.amazon.com) - https://aws.amazon.com/blogs/aws/amazon-api-gateway-build-and-run-scalable-application-backends/
- **AWS Launches API Gateway as a Cloud Service** (07-09-2015 on feedproxy.google.com) -http://feedproxy.google.com/~r/programmableweb/~3/7fyxydcfk0w/09
- **Building the TopBlogger API with LoopBack** (07-09-2015 on blog.jeffdouglas.com) - http://blog.jeffdouglas.com/2015/07/09/building-the-topblogger-api-with-loopback/

- **The New AWS API (Gateway): Anyone Who Does Not Do This, Will Be Fired. Thank You. Have A Nice Day! - Jeff Bezos** (07-08-2015 on apievangelist.com) - http://apievangelist.com/2015/07/09/the-new-aws-api-gateway-anyone-who-does-not-do-this-will-be-fired-thank-you-have-a-nice-day--jeff-bezos/
- **Create a quiz application with Google Sheets, APISpark, Streamdata.io and D3.JS** (07-07-2015 on restlet.com) -http://restlet.com/blog/2015/07/07/create-a-quiz-application-with-google-sheets-apispark-streamdata-io-and-d3-js/
- **Roll Your Own API vs. LoopBack** (07-07-2015 on blog.jeffdouglas.com) - http://blog.jeffdouglas.com/2015/07/07/roll-your-own-api-vs-loopback/
- **Screencast: Create your first API from scratch in APISpark** (07-07-2015 on restlet.com) - http://restlet.com/blog/2015/07/07/screencast-create-your-first-api-from-scratch-in-apispark/
- **Restlet Framework 2.3.3 released** (07-06-2015 on restlet.com) - http://restlet.com/blog/2015/07/06/restlet-framework-2-3-3-released/
- **Looking At API Design, Deployment, And Management From A Form Point Of View** (07-02-2015 on apievangelist.com) -http://apievangelist.com/2015/06/25/looking-at-api-design-deployment-and-management-from-a-form-point-of-view/
- **Building a self-hosted, RESTful web service with WCF** (06-30-2015 on www.codeproject.com) - http://www.codeproject.com/articles/1005498/building-a-self-hosted-restful-web-service-with-wc
- **Find a 4th of July firework near you thanks to APISpark!** (06-29-2015 on restlet.com) - http://restlet.com/blog/2015/06/29/find-a-4th-of-july-firework-near-you-thanks-to-apispark/
- **How To Easily Develop Web APIs with APISpark** (06-25-2015 on www.programmableweb.com) - http://www.programmableweb.com/news/how-to-easily-develop-web-apis-apispark/sponsored-content/2015/06/25
- **Looking At API Design, Deployment, And Management From A Form Point Of View** (06-24-2015 on apievangelist.com) -http://apievangelist.com/2015/06/25/looking-at-api-design-deployment-and-management-from-a-form-point-of-view

- **Container Apps now available in the Azure Marketplace** (06-23-2015 on azure.microsoft.com) - http://azure.microsoft.com/blog/2015/06/23/container-apps-now-available-in-the-azure-marketplace/
- **How to build your own API with the help of the OPENi API Builder** (06-22-2015 on www.openi-ict.eu) - http://www.openi-ict.eu/how-to-build-your-own-api-with-the-help-of-the-openi-api-builder/
- **How to Auto Deploy Github Repos To Heroku** (06-15-2015 on www.programmableweb.com) - http://www.programmableweb.com/news/how-to-auto-deploy-github-repos-to-heroku/elsewhere-web/2015/06/15
- **Introducing API Platform (beta): the next generation PHP web framework** (06-12-2015 on dunglas.fr) - http://dunglas.fr/2015/06/introducing-dunglass-api-platform-beta-the-next-generation-php-framework/
- **Heroku Button for Private Repos** (06-11-2015 on blog.heroku.com) - https://blog.heroku.com/archives/2015/6/11/heroku_button_for_private_repos
- **Heroku Button for Private Repos** (06-11-2015 on blog.heroku.com) - https://blog.heroku.com/archives/2015/6/11/heroku_button_for_private_repos
- **Running DreamFactory as a Docker Container** (06-11-2015 on blog.dreamfactory.com) - http://blog.dreamfactory.com/running-dreamfactory-as-a-docker-container
- **How We Moved Our API From Ruby to Go and Saved Our Sanity** (06-10-2015 on blog.parse.com) - http://blog.parse.com/learn/how-we-moved-our-api-from-ruby-to-go-and-saved-our-sanity/
- **Third Party Trade Group Releases White-Label Brokerage API For Australian Market** (06-09-2015 on www.programmableweb.com) -http://www.programmableweb.com/news/third-party-trade-group-releases-white-label-brokerage-api-australian-market/announcement/2015/06/09
- **Updated OpenShift Deploy Buttons** (06-08-2015 on blog.openshift.com) - https://blog.openshift.com/updated-openshift-deploy-buttons/
- **How to build APIs efficiently?** (05-30-2015 on api-university.com) - http://api-university.com/blog/how-to-build-apis/

- **Mertech Launches Evolution, a Multiprotocol API Dev Tool** (05-29-2015 on www.programmableweb.com) -http://www.programmableweb.com/news/mertech-launches-evolution-multiprotocol-api-dev-tool/2015/05/29
- **Building APIs with Node.js and Swagger** (05-22-2015 on www.slideshare.net) - http://www.slideshare.net/whitlockjc/building-apis-with-nodejs-and-swagger
- **20 years of Java and 10 of Restlet Framework** (05-21-2015 on restlet.com) - http://restlet.com/blog/2015/05/21/20-years-of-java-and-10-of-restlet-framework/
- **Azure APIs = Digital Legos** (05-20-2015 on www.codeproject.com) - http://www.codeproject.com/articles/993156/azure-apis-equals-digital-legos
- **The Simplest Possible ASP.NET Web API Template** (05-20-2015 on www.bizcoder.com) - http://www.bizcoder.com:80/the-simplest-possible-asp-net-web-api-template
- **Implementing bulk updates within RESTful services** (05-18-2015 on restlet.com) - http://restlet.com/blog/2015/05/18/implementing-bulk-updates-within-restful-services/
- **Versioning APIs** (05-15-2015 on blog.clearbit.com) - http://blog.clearbit.com/versioning-apis
- **Stamplay | Connect. Automate. Invent.** (05-14-2015 on stamplay.com) - https://stamplay.com/
- **Build a managed API with IBM Bluemix** (05-14-2015 on www.ibm.com) - http://www.ibm.com/developerworks/cloud/library/cl-bluemix-api-mgmt-app/index.html
- **Deeper models with composite properties in APISpark** (05-13-2015 on restlet.com) - http://restlet.com/blog/2015/05/13/deeper-models-with-composite-properties-in-apispark/
- **RESTful Day #1: Enterprise Level Application Architecture with Web API's using Entity Framework, Generic Repository Pattern and Unit of Work** (05-11-2015 on www.codeproject.com) - http://www.codeproject.com/articles/990492/restful-day-sharp-enterprise-level-application

Thanks for Tuning Into My Research!

Rembember -- You Can Find All Of This On Github (https://github.com/kinlane/api-deployment/), If You Want To Get Involved!

{"logo":"API Evangelist"}

@kinlane

@apievangelist

Make Sure And Share Your Public API Designs At The API Stack--Otherwise All Of This Won't Work!