

ESSENTIAL

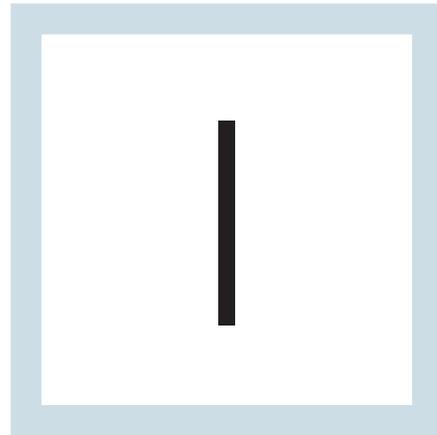
→ **Essential guide to API management and application integration**

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs



I F YOU'RE INVOLVED with integration, you have to be well-versed with application program interfaces (APIs), which are tools that make it possible to manage software applications. While invisible to end users, APIs are essential for communication between programs. Social media networking giants Facebook and Twitter are just a couple of companies that have made good use of open or public APIs. Basically, an open API is one in which an organization publishes its software to be shared freely. A startup, for example, may want to open its API, as doing so encourages third parties to use its software. This guide brings together a range of stories that highlight examples of successful API management, trends with the technology and key terms developers need to know.

[Home](#)[API development communities requires online and offline presence](#)[API design: How to properly build an application program interface](#)[Software developers demanding resource based, RESTful APIs](#)

API DEVELOPMENT COMMUNITIES REQUIRES ONLINE AND OFFLINE PRESENCE

Christine Parizo

Third-party developers aren't just uber-smart college kids coding in their dorm rooms, hopped up on Red Bull and Doritos. These days, a third-party developer could be an up-and-coming programmer or even a large company, and companies releasing APIs need to develop a third-party developer community to ensure the API's success. Companies can build API development communities in person, online and by offering incentives, according to API management-companies and veterans.

Building the developer community is critical, particularly for cloud applications, according to experts. Software as a Service (SaaS) by its nature needs to be accessible and provide benefits to users, whether or not the ideal solution comes from the API development company or a third party, said Kevin O'Brien, senior director of the AppConnect program at email marketing provider Constant Content. "The benefits we've received [from the third-party developer community] have been immediate," he said.

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

The first step, however, is to have a compelling, easy-to-use application that results in a natural fit for third-party developers to work with the API, according to John Thomas, director of product management at San Francisco-based database software provider Embarcadero. “When you build a framework that is well-designed and architected and is extensible, it means you use languages and programming patterns that make it straightforward to take the base work and [create] something additional,” he said.

For example, a framework may offer a lot of functions but not a specific one, like printing to a specific plotting printer. In that case, a third-party developer would see an opportunity to connect the framework to that type of printer, Thomas said.

GET PERSONAL -- AND OFFER PRIZES

Once the API is ready, one way to build a developer community in person is to host in-person contests like hackathons, said Alex Gaber, API evangelist at Washington, D.C.-based Layer 7 Technologies. Hackathons typically are weekend-long events dedicated to allowing third-party developers to create new ways of using a company’s API. Often, companies will offer prizes based on specific challenges, he said.

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

Another personal method to foster a third-party developer community is to run online contests for third-party developers, Gaber said. “We see big companies throw their hat into the ring,” he said, citing Samsung’s first place prize offering of \$100,000 to the developer that could build the best integration with SDK tablets, or Netflix’s movie recommendations algorithm contest that offered a \$1 million purse to the winner. “Netflix ended up with a bunch of different solutions that were built, functioned and worked,” he said.

These contests can attract more than just individual developers, according to Gaber. For example, a 20-person developer shop could submit an entry, or a large company that wants a better relationship with the contest sponsor may also throw its hat into the ring with a solution built around that company’s API, he said.

DEDICATED PORTALS PROVIDE A COMMUNITY FEELING

Meanwhile, companies looking to foster community online need dedicated portals for their third-party API developers, according to experts. “With API providers, every company in the world is going to have an API portal. These portals are where you really go to get access to these APIs,” Gaber said.

Broadsoft has a typical portal since it launched its first API four years ago,

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

complete with forums and documentation, according to Leslie Ferry, vice president of marketing at the Gaithersburg, Md.-based VoIP company. That community has grown to 5,000 members, and those members have come up with new uses for Broadsoft's API. For example, one company in New Zealand tied the API into their billing system and can send emails to remind customers that their accounts are past due, which not only is less invasive for the customer but also reduces the length of time the bill is outstanding by 50%. "Our community enables third-party developers and our own customers to create new processes," Ferry said.

Meanwhile, Constant Contact shares their insights from working with small businesses to third-party developers, alerting them to trends, according to Constant Contact's Kevin O'Brien. Utilizing newsletters, forums and webinars, Constant Contact provides developers with what they know to help the developers know where to start.

OFFER INCENTIVES AS WELL

To encourage third-party participation in the company's API development, offering incentives helps. Broadsoft offers an incubator program that provides seed funding for developers. By providing developers with money up front,

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

Broadsoft is able to speed the developers' time to create a solution for a common request, according to Ferry.

Meanwhile, Constant Contact has a revenue-sharing program in place with its third-party developers, according to O'Brien. "We have a marketplace that we promote to customers," he said. This marketplace includes the solutions third-party developers build.

No matter what, though, the developer experience should be as smooth as possible. According to O'Brien, a lesson learned is that, because of the volume of applications that a third-part developer can integrate, the developer's experience with the company's API should be straightforward and uncomplicated. "If they're making choices [about] which API to build on, it needs to reflect the functionality and be simple for them to use," he said.

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

API DESIGN: HOW TO PROPERLY BUILD AN APPLICATION PROGRAM INTERFACE

Tom Noelle

In a world of interactive componentized software, there's nothing more important than the application program interfaces (APIs) used to link components with each other, mobile devices and browsers. Build an API the right way and it helps assure functional integration and developer loyalty; do it the wrong way and compromise an entire project.

There are three ways to help get on the right side of API building:

- ▶ Know the applications and constraints of use;
- ▶ Address component architecture and binding framework; and
- ▶ Ensure changes are handled gracefully.

APIs expose features and services to developers. The way an API is used

[Home](#)[API development communities requires online and offline presence](#)[API design: How to properly build an application program interface](#)[Software developers demanding resource based, RESTful APIs](#)

and the spectrum of services represented is the primary design driver. One of the most significant mistakes developers and architects make in API building is ignoring a constituency. It's vital an API design be a good fit in the ecosystem of developers, languages and other APIs.

COMMON API DESIGN ISSUES

The REST versus SOAP debate is an example of a constraint set for APIs. Where applications already depend on one or the other, new APIs should obviously conform. Less obvious is that most APIs are part of a trend toward componentization and exposure of features. The movement may take a set of APIs more in a REST or SOAP direction over time, so be sure to anticipate that migration.

Architects can easily be bitten by conforming to object architecture and binding framework. Picking the right API design is important because it's difficult for developers to use an interface that doesn't match the architecture of the applications they're building. It should be noted that RESTful APIs typically represent resources whereas SOAP APIs represent remote processes or procedures.

Some protocol is likely used to bind APIs to API users and with Web

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

applications, which are typically Hypertext Transfer Protocol (HTTP)/HTTPS. Using HTTP with a Hypertext Markup Language or Extensible Markup Language (XML) data format, or JavaScript Object Notation (JSON) and JavaScript on client devices, makes it easy to create graphical user interfaces from an API but may not be appropriate where browser access isn't the application's intention. Some applications and APIs may use a specific Transmission Control Protocol or User Datagram Protocol port rather than Web port 80. While this can help to keep API traffic separate from Web activity, it may have firewall/security implications that will demand special system configuration that either exposes the APIs or uses them remotely.

GENERAL API DESIGN RULES

Most APIs can be viewed as syntax of verbs and nouns. For example, a sentence with a verb representing a requested action (get, put, delete) and the nouns signify arguments appropriate to the action. It's good practice to always generate a status/result variable that communicates error conditions or successful execution. The error conditions should be comprehensive enough to communicate problems unambiguously.

The semantics of the API, meaning the syntax of the functions provided, are

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

important because the ability of the API to convey its services and parameters clearly reduces developer errors. One critical point is that if the API represents a stateful service, the function semantics should be session-oriented (find-record, update-record, delete-record) where the stateful nature of the service is made clear.

It follows that if, as in this example, the update and delete functions operate on the previously located data element, then the update and delete functions do not supply their own data element keys; they'd be redundant and risk generating developer confusion. On the other hand, a stateless service must always provide all of the data since no session context could be inferred.

COMMON QUESTIONS AND ISSUES

Syntax issues created by updates or changes to the API are often ignored. There are two sides to each API and the change process can pull them out of synchrony. Some architects will express a version variable in an API to ensure that both sides are expecting the same formats. At minimum, both the server and client side of an API should perform basic validation to protect against changes that create a mismatch of syntax so they don't contaminate information or crash applications.

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs

Another common question has to do with the data format. XML is the most general way of expressing parameters and exchanging information, and it is applicable to REST and SOAP interfaces. But XML processing is a heavy-weight and most valuable for expressing structureless data. For REST, JSON has gained favor as being easier to use while providing some specific variable typing widely used and expected in API building. Where APIs exchange rigidly defined data elements, JSON is likely a better choice for RESTful exchanges.

API testing is often lumped into application lifecycle management processes. Some of it rightfully belongs there, but there should also be specific unit-test processes designed to validate APIs and establish that they'll perform gracefully even when data includes errors. The looser the data binding and typing of an API, the more risk there is of passing information that will result in a later error or crash. This is why it's important to adopt tight constraints on variables and to test each API with a range of data.

API build problems will destroy an application faster and more thoroughly than almost any other type of architecting error. Spending more time designing APIs to anticipate current error conditions and future changes will be time well spent.

[Home](#)[API development communities requires online and offline presence](#)[API design: How to properly build an application program interface](#)[Software developers demanding resource based, RESTful APIs](#)

SOFTWARE DEVELOPERS DEMANDING RESOURCE BASED, RESTFUL APIS

Jason Tee

Well seasoned enterprise architects know that there is no silver bullet in the world of IT, and no single solution will ever meet every requirement criteria to the tee, but when it comes to web service development, a REST based approach is becoming more and more popular. REST may not be the best approach for every situation, but it does have enough virtues to put it on the short list for any organizations that is comparing the various different approaches to web service development . And the fact is, a REST based approach can be effective in even the most challenging corner-cases. The key is that developers simply have to use REST the right way.

THE EFFECTIVE APPLICATION OF RESTFUL TECHNIQUES

Needless to say, using a REST based approach for developing a distributed architecture effectively becomes a whole lot easier when there is a proven and effective structure within which applications can be built. In the world of

[Home](#)[API development communities requires online and offline presence](#)[API design: How to properly build an application program interface](#)[Software developers demanding resource based, RESTful APIs](#)

distributed computing, if you want to understand RESTful development, you scour the globe to find an expert. In this case, TheServerSide caught up with Jerome Louvel (@jlouvel), coauthor of Restlet in Action and the creator of the Restlet framework. Lovell shared several insightful tips for getting the most out of RESTful web development.

One issue web developers often raise is how to get things done with REST when it comes to handling complex, real world transactions that involve multiple resources whose interactions need to be coordinated. Louvel addresses the topic of e-commerce by framing it as a simple use case with a straightforward solution. For example, to make a traditionally developed online shopping application into a RESTful experience, an architect simply needs to break down the associated business transactions into all of their underlying components.

In a transactional, online e-commerce type of system, people need to be able to browse products, select products, view and edit their cart, enter payment and shipping information and then make a final decision to pay. At any point up to the point where the user completes their checkout, users need to be able to navigate back to a previous step. From a pure API standpoint, this means mapping all the important parts of the transaction into resources via URIs. At the end, the transaction is completed by sending a final HTTP request that

[Home](#)[API development communities requires online and offline presence](#)[API design: How to properly build an application program interface](#)[Software developers demanding resource based, RESTful APIs](#)

coordinates the state of each of the resources involved in the transaction. “It’s just viewing your transaction as a set of one or multiple resources depending on the complexity of your business processes. Technically, nothing prevents you from doing that. It’s just not trying to map what you’re used to doing with RPC or putting too much context in one request,” says Louvel.

EASING INTO A RESOURCE BASED APPROACH TO APIS

Designing a RESTful API is often about unlearning what you knew before. Developers run into problems when they try to overlay their previous knowledge and understanding of distributed computing with RPCs. That’s when developers find themselves creating more problems than they fix. Louvel’s advice is not to take the usual, traditional approach when developing RESTful APIs. Forget what you learned from the service-oriented, RPC based world. Then, you’ll discover that you are working without limitations and you can really start to solve your business problems. Furthermore, the modularity involved in developing a RESTful architecture means developers have smaller, more agile parts that provide greater flexibility when designing solutions that really work.

The main takeaway Louvel has for Java developers is this: apply as many principle from the tenants of RESTful development and design as possible,

[Home](#)[API development communities requires online and offline presence](#)[API design: How to properly build an application program interface](#)[Software developers demanding resource based, RESTful APIs](#)

but at the same time, use only as much as is needed. See everything as a resource. Use calls from the HTTP protocol, especially GETs, PUTs, POSTs and DELETEs to effectively create and update your resources. Eventually, approaching the problem domain in a RESTful manner becomes second hand. By applying these foundational concepts, developers get a lot of built-in features that are common to the world-wide-web such as caching and automatic compression for free.

And once the development team has got the basics down, kick it up a notch and try to see every problem as having a RESTful solution. Says Louvel, “try to use hypermedia as much as possible to become the engine of your application. This is a principle that is difficult to apply and people are still trying to find the best way to do that.”

Home

API development communities requires online and offline presence

API design: How to properly build an application program interface

Software developers demanding resource based, RESTful APIs



FREE RESOURCES FOR TECHNOLOGY PROFESSIONALS

TechTarget publishes targeted technology media that address your need for information and resources for researching products, developing strategy and making cost-effective purchase decisions. Our network of technology-specific Web sites gives you access to industry experts, independent content and analysis and the Web's largest library of vendor-provided white papers, webcasts, podcasts, videos, virtual trade shows, research reports and more —drawing on the rich R&D resources of technology providers to address market trends, challenges and solutions. Our live events and virtual seminars give you access to vendor neutral, expert commentary and advice on the issues and challenges you face daily. Our social community IT Knowledge Exchange allows you to share real world information in real time with peers and experts.

WHAT MAKES TECHTARGET UNIQUE?

TechTarget is squarely focused on the enterprise IT space. Our team of editors and network of industry experts provide the richest, most relevant content to IT professionals and management. We leverage the immediacy of the Web, the networking and face-to-face opportunities of events and virtual events, and the ability to interact with peers—all to create compelling and actionable information for enterprise IT professionals across all industries and markets.