

The **7** habits of highly effective API and service management



The 7 habits of highly effective API and service management

A New Enterprise challenge has emerged. With the number of APIs growing rapidly, managing them one-off or via Excel or corporate wiki is no longer feasible.

The smartest organizations have discovered a set of best practices to design powerful APIs that leverage existing services, to effectively manage those APIs throughout their lifecycle and to scale their deployment across consumers and devices. This eBook examines the relationship between APIs and services and presents the key elements of a successful API strategy in the form of 7 habits.

In this eBook you will learn:

- How to leverage existing services in the API economy
- Where to get started with your API strategy
- Key criteria for selecting an API Management solution
- Strategies to overcome API security and identity challenges
- How and why to apply the fundamentals of API First Design

Table of Contents

Changing the game: A shift in strategy

Next gen SOA: API management

What is an API

Driving forces behind API adoption

The 7 habits of highly effective API and service management

Habit 1 Apply an API-first design approach

Habit 2 Choose a solid API runtime

Habit 3 Create a central service repository

Habit 4 Manage services through versions, policies, and contracts

Habit 5 Promote and socialize your APIs

Habit 6 Monitor and assess API usage

Habit 7 Continually improve

Conclusion

Changing the game: A shift in strategy

Service Oriented Architecture sets the stage for API management

Over the last decade, many companies implemented SOA to one degree or another in the name of reuse and consistency. By decomposing functionality into atomic services, SOA promised to deliver streamlined processes, increased efficiency, and lower costs. However, few SOA initiatives delivered on this promise.

SOA had the right ambition but the wrong approach. The original intent of SOA – to enable reuse of services, provide flexibility and scalability and to allow interoperability between services and applications – was exactly what businesses wanted. The problem however, was in execution. Organizations incorrectly structured their SOA projects with a “top-down” approach that caused numerous issues across the enterprise. This top-down approach dictated a rip-and-replace strategy whose aim was to gradually replace aging enterprise systems with modern alternatives. The problem was, these projects often dragged on for years, consuming far more resources and budget than planned, ultimately becoming an expensive burden to the business.

This is one of the main reasons why, for many, SOA has become a relic of the past, a word replete with negative connotation. The monolithic, top-down initiatives of the past have failed, costing businesses millions of dollars and ultimately failing to deliver projected business value. SOA was essentially considered dead. Nevertheless, the fundamental principles of SOA are still sound and have been given new life in the form of API management.

Next gen SOA: API management

API management is essentially the next generation of SOA. But how will API management avoid the well-known pitfalls that derailed so many SOA initiatives? While API management shares many commonalities with SOA, there are two key differences. First, rather than a top-down, rip-and-replace approach, API management uses a “bottom-up,” wrap-and-renew approach. APIs essentially breathe new life into services by layering accessible and developer-friendly interfaces on top of them, effectively shielding consumers from the underlying technology. Secondly, API initiatives are all about ROI. Projects have defined milestones and rapid implementations, making impact immediate and ROI easy to measure.

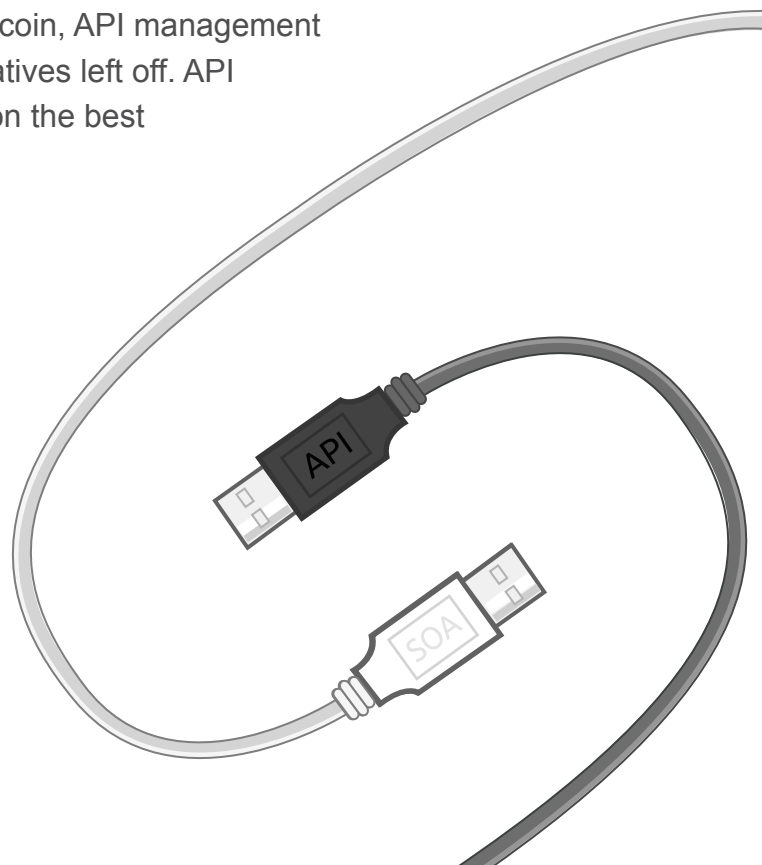
This approach delivers on the initial promises of SOA and more:

- Accelerated application development
- Reduced risk of failure
- Improved developer effectiveness
- Agility and flexibility

Often considered two sides of the same coin, API management has emerged to pick up where SOA initiatives left off. API management adopted and improved upon the best of the underlying principles of SOA.

SOA and API management share key core principles:

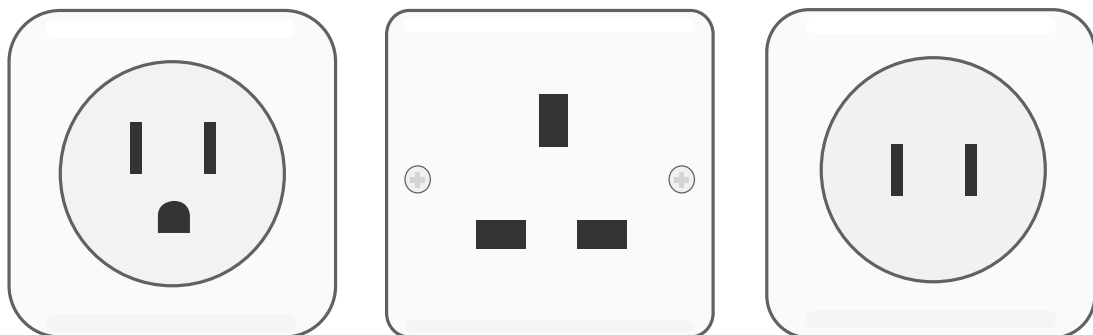
- Abstraction of business logic
- Solid governance frameworks
- Reuse of services



What is an API?

APIs can be defined in multiple ways. In fact, some think that API and a service are interchangeable terms, while others think they're at opposite ends of the spectrum. Before moving on, it's important to establish a common definition of API. In simplified terms, an Application Programming Interface provides a way for developers to interact with and consume a service.

An analogy is useful in illustrating what this really means, so let's consider a service that all of us consume every day - electricity. Electricity is delivered to consumers by a utility company. The utility company provides a service (electricity) to consumers that is accessible through electrical sockets. These sockets vary from one country to another, limiting access to only those consumers with the correct plugs for that socket. The plugs are essentially consumers, only able to make use of the service if they have the appropriate authorization. In this case, the API is the socket itself. APIs are gateways between services and consumers, providing consumers access to services through various interfaces depending on their credentials. In addition, consumers can utilize the services they receive and implement them in their own way. Consider a laptop, which consumes electricity through a socket. Through its own "API," a USB socket, it can provide that same electricity service to charge other devices.



Driving forces behind API adoption

“ Gartner predicts that the full life cycle API management market will reach \$1.6 billion by 2021.

Organizations are beginning to understand the importance of APIs and the value they deliver to business. But what exactly is causing all the fuss?

Business agility: Much more effective than creating business logic and exposing it as a website is creating business logic and exposing it as an API. With APIs, it doesn't matter how the logic is used; clients and consumers can consume the API and expose it to their liking, delivering efficiency and enabling innovation for the enterprise.

API economy: Businesses are developing “API products” as new sources of revenue. Expedia generates over \$2 billion annually through the data made available through their API. Salesforce generates 75% of their revenue through API activity. The API economy is here and growing fast.

Internet of Things: The number of things that can be connected is growing fast; everything from your coffee pot to your thermostat to your car can now be accessed via API. As a result, businesses are finding more and more opportunities to create connectivity in new places.

Now that we've defined what APIs are, their relationship to services and SOA, and the reasons behind their growing importance in the enterprise, it's time to dig into the 7 habits that drive effective API management. If you're starting an API initiative or wondering how to get started, following these best practices will ensure that your APIs are easy to use, fast to deploy, and deliver the business results you expect.

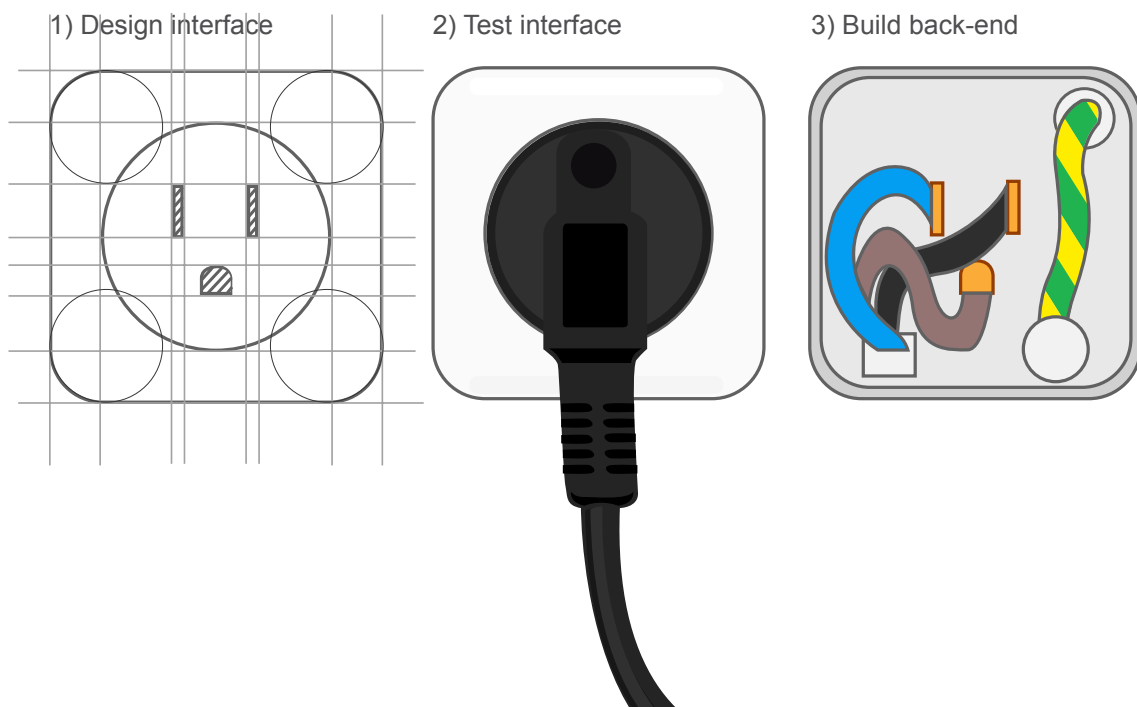
Habit 1:

Apply an API-first design approach

Typically, a business owner starts off by identifying the need for an API. Next, the business owner works with an architect to actually design the API and from that comes the specifications of the API. Once the specifications are complete, they are communicated by the architect to the API development team that will implement the back-end of the API.

With an API-First approach, rather than implementing an application and then building an API on top of it, businesses first create the interface and then put the back-end logic into place - whether it be cloud-based or on-premise.

This method allows organizations to isolate concerns and to focus on clear, well-defined API specifications instead of implementation concerns. API-First development provides a simple framework that enforces better REST API design practices, helping businesses focus on API resources and messaging before undergoing the heavy lifting and back-end implementation.



Habit 2:

Choose a solid API runtime

Once an API is designed, created and its back-end implemented, it's time to find the right runtime. Choosing a runtime is crucial, as it affects how successful your API strategy will be in terms of service, liability, scale and ability to meet future needs. So what key capabilities should you look for in an API runtime?



Hybrid support: As more and more businesses are moving to the cloud, it is important to find a runtime that allows you to deploy applications both in the cloud as well as on-premise without having to modify anything. This allows businesses to burst into the cloud when extra resources are needed or even shift from on-premise to the cloud if and when your business needs require. The ability to develop applications once and deploy them in the cloud or on-premise provides a host of possibilities without complexity.



Scalability, reliability, availability: These “-ilities”, along with performance, are crucial when searching for a solid API runtime. The last thing you want is an unstable runtime that crashes or experiences outages. Choosing the right enterprise grade technology for your API runtime is crucial to the success of your API.



Strong Orchestration: A successful runtime should provide strong orchestration and orchestration capabilities. The ability to carry out complex back-end orchestration plays a key role in translation between the API layer and back-end implementation.

Habit 3:

Create a central service repository

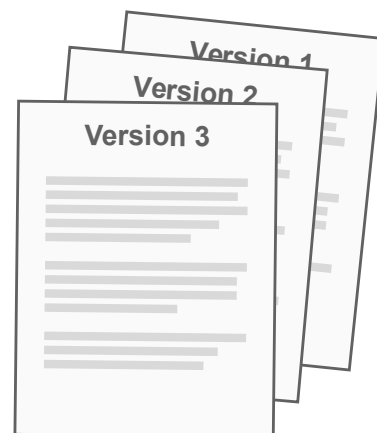
After designing, developing, and running the API on a solid platform, exposing the API in a central repository is paramount. Surfacing the API facilitates API discoverability and accessibility by people who need it. A central service repository also makes it easy to categorize and search through services while providing a consolidated view of APIs; it is the place for all design and runtime governance.

Forums and documentation make it easy for developers to assess an API's fit for purpose and to get started quickly with API consumption. Finally, with visibility into key metrics of APIs, business owners and architects know how an API is performing so they can take corrective action and continue to improve the API in future versions.

Habit 4:

Manage services through versions, policies, and contracts

Tracking service versions and consumers gives businesses insight into who is using APIs, which versions they're using and how they're using them. This helps with API lifecycle management and allows API publishers to assess the impact of retiring a version. In addition, policies and contracts are crucial to enforcing security and managing SLAs with API consumers. An API solution should offer a way to easily create well-defined policies and contracts and associate them with the right APIs and consumers.



Habit 5:

Promote and socialize your APIs

Creating a developer portal to establish a community around your API is important to its success. By making it easy for users to follow your API, download documentation and ask questions, API publishers can engage with API consumers on an ongoing basis. Providing and promoting content that simplifies API consumption strengthens your user base, ultimately helping your API be successful.

Habit 6:

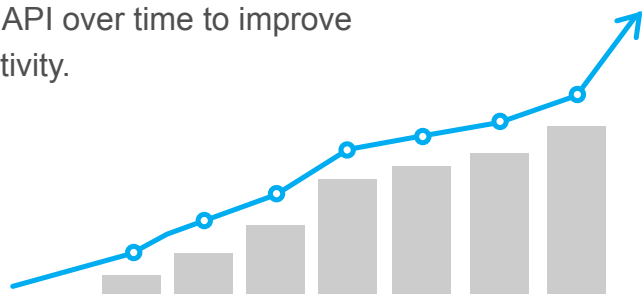
Monitor and assess API usage

Understanding which parts of your service are being used is important, but it's only part of the picture. You need to see how consumers are using it. With metrics for both overall usage and per consumer usage, businesses can closely monitor API activity and engagement. The ability to monitor API usage over a defined period of time and to understand usage patterns from both technical and business perspectives is valuable as it helps business owners and technical teams better understand their users and ultimately create a better service.

Habit 7:

Continually improve

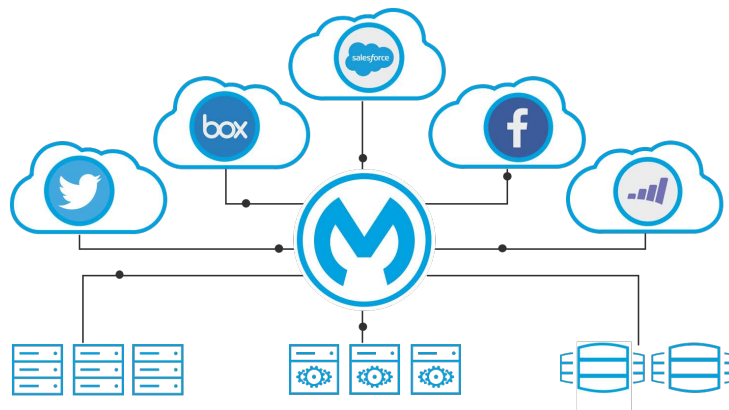
The ability to refactor your APIs by iterating through habits 1-6 multiple times allows you to optimize your API over time to improve consumer experience and productivity.



Conclusion

With the progression towards a more connected and API-driven world, defining the right API strategy and selecting the right API platform is crucial to supporting innovation and growth. Following the 7 habits outlined above will ensure a winning API strategy. To make that strategy a reality, MuleSoft's Anypoint Platform for APIs delivers an end-to-end solution to design APIs that developers will love; easily connect them with backend services; run them in a secure, scalable environment; and manage them throughout their lifecycle.

Learn more about MuleSoft or contact us to discuss your API or integration needs today.



About MuleSoft

MuleSoft's mission is to help organizations change and innovate faster by making it easy to connect the world's applications, data and devices. With its API-led approach to connectivity, MuleSoft's market-leading Anypoint Platform™ is enabling over 1,000 organizations in more than 60 countries to build application networks. For more information, visit mulesoft.com.

MuleSoft is a registered trademark of MuleSoft, Inc. All other marks are those of respective owners

