



DIGITALLY REMASTERED

APIs Connect Your Digital Nervous System

When every business—be it a startup or stalwart—is connecting with customers through software, it becomes imperative to deliver new experiences and services quickly. While incorporating software development as an integral part of business may seem like a daunting task, the first important step is to recognize the fundamentally different nature of software’s role in business today. This isn’t simply a matter of traditional IT with some added responsibility. The ability to deliver customer-facing, software-based experiences is an entirely new capability that you may need to build from the ground up. One way to accelerate your time to market and open new opportunities is to leverage APIs to connect systems and data. APIs are critical to powering today’s complex apps, which rely on multiple data streams and third-party services to operate.

In this excerpt from “Digitally Remastered,” you’ll learn the importance of APIs to your digital transformation initiatives and how to best harness their power.

The API Advantage

The value of APIs is not limited to pure technology players alone. The app economy opens the playing field to all businesses who can leverage APIs to fuel a wide variety of initiatives across a diverse set of industries. The leaders of tomorrow will be companies that embrace web APIs to create superior new integrated solutions and experiences for their customers. Research from Freeform Dynamics, an independent IT research and analysis firm, stresses this broad impact and reach: enterprises with advanced API implementations are 2.6 times more likely to have faster delivery of revenue-enhancing apps and 2.3 times more likely to have lower IT-related costs and risks.³ A sound API strategy serves a host of possible business objectives — from acceleration of app development to driving consistent user experiences to accessing new channels.

Acceleration of application development

A primary benefit of APIs is the sheer speed at which new applications and features can be built and delivered. This benefit extends beyond your own APIs; a wealth of third-party APIs can help you build your software experiences without having to reinvent the wheel. Need a map in your app? You can use the Google Maps API. Facebook social integration? No problem. Credit card processing? There are services with readily available APIs to handle that. Twitter, YouTube, Flickr, Spotify and many more all offer web APIs to enrich the user experiences you deliver with social and online media.

Even if you're not a programmer, using APIs to perform basic Facebook integration is a snap, as the figure below makes clear.

Graph API Call

Let's make an API call to publish a message. Add the code into the response function of the `FB.login` call you added above:

```
FB.login(function() {  
  // Note: the call will only work if you accept the permission request.  
  FB.api('/me/feed', 'post', {message: 'Hello, world!'});  
}, {scope: 'publish_actions'});
```

Try the script. A status message will be posted to your timeline:



The speed at which APIs can create new value gives your organization the ability to respond quickly to market changes and new customer demand.

Delivery of quality multi-channel experiences

APIs give your organization the ability to connect systems and share data to deliver a consistently high-quality customer experience across multiple channels. Although the target device platforms themselves may be very different, the underlying APIs that feed them with data and services to power the experiences you deliver can be engineered to be uniform and consistent. This API-level consistency helps to ensure consistency and quality when it comes to the user experience. The device-agnostic nature of web APIs is what makes them such a powerful tool for aggregating and composing functionality in a uniform way even across a very diverse set of devices. For example, suppose you need to display images of your products on apps that run on phones, tablets, PCs, and game consoles. One option would be to implement a separate API for each device type. This approach would create needless complexity that would increase the potential for errors and the possibility that product images would not be consistent across devices. Instead, you can implement a single API to provide the appropriate image for each device. This approach reduces overall complexity and the chance of errors or unintended inconsistencies, and helps ensure a seamless, high-quality multi-channel user experience.

Access to new business opportunities

APIs are a powerful tool for creating new business opportunities. They allow you to expose valuable data and information that you already have in order to build new products and services, or to enable a partner-based ecosystem around your own business. APIs allow you to take a database — or to aggregate data from multiple data sources — and expose the valuable information they contain programmatically as live services. Such API services can form the basis of broadening access to your business and the value that it provides. For example, Weather Underground offers a tiered pricing structure for its APIs and targets them directly to developers looking to enhance their apps with weather-based information. And Sonos offers third-party developers access to their APIs in order to bring more value to their devices by enabling an app ecosystem through their API platform.

API Challenges

APIs are an essential ingredient in your digital transformation and they can rapidly expand the reach and potential of your business. As with any major software initiative, however, there are a number of important areas to address as you plan and implement your API strategy.

Design

A common mistake with APIs is thinking: “We have all this data we’re going to expose with APIs.” An ad-hoc approach to API creation will lead to wasted effort and a jumble of APIs that don’t get used or can’t be used efficiently. The APIs you build must focus on enabling

new features and scenarios. The design of your APIs must also consider the needs of developers — the users of APIs. APIs that are difficult to use or error-prone can cause delays, needless effort, reduced adoption, and compromised application quality. APIs should make the job of developers as easy as possible by hiding the maximum amount of work and complexity behind the API implementation. As a simple example, consider an API that returns the results of a database query in a product catalog. Suppose that the query may produce a number of unhelpful redundant results — identical copies of the same product in the list. One way to design the API would be to leave those redundant results in place for the API user — the developer — to deal with. A better API design removes the duplicate items in the list before returning the results. Doing so reduces the amount of work every developer using the API must do, and decreases the chances of unintended consequences or side effects.

Integration

The existence of multiple API methodologies and data formats can also cause some headaches. Depending on the requirements of your API users, you will need to develop APIs that support one or perhaps even both of the two prevalent data exchange formats (JSON, short for JavaScript Object Notation, or XML, which stands for eXtensible Markup Language). Although REST (Representational State Transfer) is the most common method of accessing APIs, other methods may be more appropriate based on the specific use. For example, MQTT (Message Queuing Telemetry Transport) is a common interface for IoT and some mobile applications, and SOAP (Simple Object Access Protocol) also addresses specific use cases. In addition to being thoughtful about API protocols in the design of your APIs, your development team will also need to be adept at using APIs external to your organization, since they will be using a variety of API protocols that may be different than your own.

Security

APIs are not inherently secure. By their very nature, APIs provide access to often highly sensitive data at the core of your business and magnify the dangers of any lapse in security. It is critical to ensure that your APIs don't compromise your sensitive data, and that only the right set of users, apps, and devices can access your APIs based on the appropriate controls. You must integrate security deeply into your APIs with rigorous design and security measures such as encryption, robust authentication, and risk management. There are many examples of security lapses; some are as basic as a username and password hard-coded — embedded directly — into the API itself. In another instance, a non-secure API was used to download all the usernames associated with an online service; these were then used to attack other popular services. You don't want to make the news this way!

Scalability and performance

Consistently high performance of your APIs is critical to delivering a great user experience. We've all encountered — and been frustrated by — crashes, hangs, or lack of responsiveness trying to access an app or online service. The cause is often an API failing to perform. Popularity of your service and the APIs that power it is a good thing, but you have to ensure it can scale. You must architect your APIs and all of the software and systems supporting them so that anticipated usage won't be overwhelming when traffic increases. Your APIs should also be designed to gracefully handle variable network speeds and latency. For example, your API can be designed to

provide lower-resolution and more space-efficient images as needed to decrease the amount of network traffic — and time — required for the user to see images on their display. Or your API can return a batch of data in a specified range to decrease unnecessary back-and-forth between the app and the API that can be very time-consuming when dealing with poor network responsiveness.

Discoverability

It can be difficult to find the right API for the right task, and your APIs will be no exception. APIs are not a “build-it-and-they-will-come” technology. If you want people — third-party developers, partners, and your own developers — to use your APIs, you need to get the word out and document how to use them. This should include sample code that can serve as a helpful starting point for developers learning to use your APIs.

Lifecycle management

Creating a new API is just the beginning of a lifecycle through which the API will evolve, mature, and ultimately reach the end of its usefulness. Building a new API is a long-term commitment, and you should strive to have the fewest number of APIs possible. It is all too easy to build a proliferation of APIs, causing a confusing jumble of functionality and a support and maintenance nightmare. You must include the evolution of your APIs as part of your design process to ensure that you build in both forward and backward capability. Future versions of your APIs must continue to support apps using previous versions, and you have to leave enough flexibility in your initial designs to allow API evolution and the addition of new capabilities to future versions of your APIs. As your organization’s API savvy and use grows, you need a system in place in order to track API versions, gracefully sunset old or outdated APIs, and ensure the latest API versions are adopted as quickly as possible. A forgotten older API could be an unintended backdoor into your systems, and you also have to be mindful to maintain backward compatibility as needed to ensure that your partners’ and your own older applications continue to operate.

Getting Started

APIs are an essential tool for delivering new value to customers and partners quickly and at scale. A few critical considerations will help your efforts deliver on their full potential:

Invest based on value. Just because you can build an API doesn’t mean that you should. Avoid the temptation to build new APIs simply in response to their popularity; choose your API investments based on their value to your customers and your business. Carefully evaluate the use cases for new APIs — or enhancements to existing ones — to ensure that they are clear, compelling, and provide lasting value. Any new API you develop should have at least one initial user to ensure that the API is fulfilling a concrete and well-defined need.

Plan and build for the long run. It’s important for you to take the long view in building out your API program. APIs that you build and deploy today may be in use for many years. Architect your APIs carefully to be able to adapt to the changing needs of your business.

Maintain the smallest set of APIs possible to minimize development overhead and needless complexity, and keep in mind that every API you add to your portfolio has an ongoing cost. APIs are a foundational technology; you must ensure that your APIs are designed and built with sound architectural principles at a high level of execution quality. Any deficiencies in the design and implementation of your APIs will be magnified as their use increases; make sure that you quickly address any shortcomings and are constantly improving the quality of your API technology.

Focus on your developers. Developers are the target audience of your APIs. Just as you must consider the value that an API delivers to customers, you must also consider the developer experience with your APIs. Your goal must include maximizing developer productivity. This includes high-quality design and execution of your APIs and high-quality sample code and documentation to accelerate adoption and increase the effectiveness of your APIs. You must be ready to support your APIs with problem reporting, tracking, and resolution. Your APIs should do as much work as possible to remove as much complexity and overhead from the developer as possible. Building the needs of the developer into your API program will help ensure both development velocity and product quality.

Scalability, security, and performance are essential. A well-designed, well-supported API portfolio is just part of the API picture. Your APIs must also deliver consistently secure, reliable, high performance operation. Security must be a core feature of every API you build to ensure that you never compromise your customers' trust. You must ensure that only authorized users have access to your APIs, that you protect sensitive data and privacy, and that you respond quickly to new threats. Performance and reliability are also at the heart of a high quality user experience. You must design and implement your APIs to provide consistently high performance and to be able to quickly scale to accommodate use.

Powering L'Oreal's digital transformation through APIs¹

L'Oreal began embracing web APIs two years ago to accelerate its digital transformation. The company needed technology designed to be sharable, flexible, nimble and scalable — enabling L'Oreal to create the personalized digital services today's connected consumers expect.

“We wanted to up our digital game at L'Oreal. The L'Oreal philosophy is to be very, very forward looking. APIs are really the way of doing business for us right now,” comments Susannah Greenberg, Vice President, Technology at L'Oreal.

At L'Oreal, APIs improve partner communication and development because they enable backend systems to more quickly and efficiently share pricing data and product availability across vendors, sites, L'Oreal brands, and platforms.

APIs provide agility and cost advantages, too. The company can tap into partners' merchandising data using APIs. Having an API gateway and well-defined abstraction of API services means L'Oreal can switch between third-party service providers (such as an email marketing provider)

or adopt new capabilities faster. As a result, APIs have enabled L’Oreal to support a fivefold increase in the number of endpoints with partners and vendors without expanding its internal teams.

Today, nearly a dozen APIs now power some of L’Oreal’s key consumer services, enabling a variety of features. They include front-and-center beauty apps, such as Makeup Genius, which uses facial recognition technology to let shoppers virtually try on 420,000 products from lipstick to eyeliner. By combining their internally developed APIs with third-party APIs, L’Oreal has been able to create unique and compelling digital experiences that deepen the company’s engagement with its customers.

Learn more:

See how CA can help you create and manage your API portfolio.
ca.com/api

Get the entire book:

Download the entire Digitally Remastered book to learn how to fully tool your software factory and drive successful digital transformation.

1 CA Technologies Infographic, "APIs: The Building Blocks of the App Economy," 2016, <http://paidpost.nytimes.com/ca-technologies/apis-the-building-blocks-of-the-app-economy.html>