# **What is an API?**

* An API (Application Programming Interface) is a set of functions that allows applications to access data and interact with external software components, operating systems, or microservices.
* To simplify, an API delivers a user response to a system and sends the system’s response back to a user. You click “add to cart;” an API tells the site you added a product to your cart; the website puts the product in your cart, and your cart is updated.
* You may hear the term “microservices” come up in relation to API. These however, are not the same. Microservices are a style or architecture which divides functionality within a web application. While API is the framework which developers interact with a web application. Microservices can actually use API to communicate between each other.
* API lets a developer make a specific “call” or “request” in order to send or receive information. This communication is done using a programming language called “JSON.”  It can also be used to make a defined action such as updating or deleting data. There are four basic request methods that can be made with API:

1. GET – Gathers information (Pulling all Coupon Codes)
2. PUT – Updates pieces of data (Updating Product pricing)
3. POST – Creates (Creating a new Product Category)
4. DELETE – (Deleting a blog post)

# **What is JSON and why is it used?**

JSON (JavaScript Object Notation) is used to represent data on a server. It’s fairly easy to read by humans, and easy for machines/applications to understand. Let’s look at an example of JSON from a product on BigCommerce:



This is easy to understand as it’s outputted in key/value pairs, with the key on the left, and a value on the right. Keys are a fixed object defined by the application and will remain the same as with “category.” Whereas the values will be unique, such as “Shirts.”

# **What is an API Request?**

There are several components of an API Request in order for it to function. Let’s go over these individually and how they can be used to build a request.

**Endpoint**

There are two key parts to an endpoint that are used when making an API request. One of which is the URL. BigCommerce uses https://api.bigcommerce.com/stores/ as the URL for all API Requests. This may look like a regular URL but if you plug this into a web browser, you will receive a 404 error message.

The second part is the path. The path will vary depending on what you are trying to accomplish. You can find a list of available paths for BigCommerce by visiting our developer documentation: <https://developer.bigcommerce.com/api-reference>. For this example we are going to use the product path which is /v3/catalog/products.

When we put these two parts together, we get a complete endpoint [https://api.bigcommerce.com/stores/{store\_hash}/v3/catalog/products](https://api.bigcommerce.com/stores/%7B$$.env.store_hash%7D/v3/catalog/products). Now you may be saying to yourself “What is the {store\_hash}?“ “Where did that come from?” This is what is known as a variable. Variables are unique components to an endpoint and will vary depending on your store’s information. You can spot a variable by the open and closed brackets “{ }”.

**Header**

Headers provide information to the client and server. Common examples of a header would be authentication credentials such as a “Auth Token” or “Client ID”. These credentials are provided to you automatically when you create an API Account. Another common header is referred to as the “Content Type,” which informs the server about what type of content will be sent. For example, a commonly used content type is  “application/json” which let’s the server know, we are sending JSON data across.

**Method**

Methods are the actions taken when sending a request. Think back to the beginning when we discussed GET, PUT, POST, and DELETE. These are all API Methods.

**Data**

The request data, also commonly referred to as the “body,” is information that will be either sent to or returned by a server. In the previous discussion of JSON, you can see an example of API data. The body of a request will sometimes require specific information before it can be delivered. An example of this is if you are editing a single product, the Product ID will be required before any change can be made.

# **What about REST & SOAP APIs?**

While API follows a specific set of rules that determine how programs communicate with one another. REST & SOAP define how the API is presented. Each are similar in functionality but have several key differences and use cases.

REST stands for “Representational State Transfer” and is the set of rules that developers follow when creating an API. REST is read using JSON as we covered previously. One of these rules is that the API should be designed in a way that is easy to use and will make sense for developers. An example of not following this rule would be to have the product endpoint “prod\_839” instead of just “products.” As this could cause the API to be fairly unpleasant to work with.

SOAP or Simple Object Access Protocol is another design modal for web services. Instead of the typical JSON that REST API uses. SOAP uses a language known as Extensible Markup Language (XML). XML is designed to be machine- and human-readable. SOAP follows a strong standard of rules, such as messaging structure and convention for providing request or responses.

This article in general refers to REST API standards as it’s mostly used in today’s world. Major companies such as Google, Amazon, eBay, and even BigCommerce use REST APIs. REST is generally the preferred choice amongst software developers.

# **Everyday Examples of APIs**

API helps developers quickly deliver information to consumers and is used every day in today’s world. From [shopping online](https://www.bigcommerce.com/articles/ecommerce/), browsing a social media app, or playing a game on your smartphone. Every time you visit a page online, you’re interacting with API. Here are some real-world examples of how you interact with API and may not even realize.

Going to a bank.

Think of yourself as a user and a bank teller as an API, while the bank is the system you interact with. When you want to take some money out of your account, you walk up to the teller(API) and say “I’d like $1,000 from this account”. The teller (API) then goes to the back, tells the bank manager (the system) “Mr/Ms.X would like $1,000”, the bank manager (the system) gives the teller (API) $1,000 who eventually gives it to you. As you can see the API, is a messenger between your needs and the system.

Searching for hotels.

When you go onto a travel site, it may be linked to 10 other travel sites to find the best deal for you. When you input details like Atlanta, 2 nights, 1 Room, you send this request to those 10 different travel sites. The API takes your request for that specific location, date range, and room and pings the 10 sites, who send back the deals they have found. You look through the 10 deals and pick the best one. Again, the API is a messenger for your requests.

Finding a Facebook profile.

Stalking your ex? Hopefully not, but thanks to APIs, you can do it easily! If you type in “John Smith” on Facebook, the API informs Facebook’s servers that you’re looking for John Smith. Facebook then sends you a list of all the profiles that match that name (with factors like vicinity to you, or mutual friends). Now you can find John Smith!

Finding a new restaurant.

Let’s say you are traveling to a new city or state. You’ve just dropped everything off at the hotel and decide to grab some lunch. You grab your smart phone and look up restaurants nearby. Quickly you are shown dozens of local restaurants right outside your hotel. Thanks to Google Maps API; they are able to easily display business hours, reviews, phone numbers, and even times they are likely to be busy.

Staying up to date on social media.

You’re stuck in a cab while in rush hour traffic. The worst, I know! You decide to kill some time and decide to catch up on what’s happening in the world of sports. You open Twitter and navigate to the ‘Sports’ section. Twitter’s API allows you to easily see various tweets relating to your favorite team winning the play-offs. You’re even able rub it in your friends face that his team lost by retweeting the final score. From here Twitter knows to take this tweet, and display it to everyone who follows you.

# Why Modern Ecommerce Sites Use APIs

API’s offer a wide range of benefits for [Ecommerce Sites](https://www.bigcommerce.com/blog/?p=16686). They can help consumers easily find products, grow a company’s brand, or even expand their earning potential by selling products on various marketplaces such as [eBay](https://www.bigcommerce.com/blog/selling-on-ebay-for-beginners/), [Amazon](https://www.bigcommerce.com/blog/selling-amazon-bigcommerce/), and Facebook. Listed below are some benefits of why API is so important to ecommerce sites today.

Security.

Security is enhanced when sites use APIs. Whenever you send a request, you aren’t directly linked to a server. You send small amounts of information, the API delivers it, and the server sends it back. This minimizes the risk of a breach or someone accessing the backend of a server.

Speed.

Without APIs, you would have to call a store and ask them to look at their inventory from all their suppliers, which they would eventually get back to you. This, instead of having an API where you could easily see what a product was, the price, or it’s stock level.

Scalability.

APIs allow scalability and flexibility when expanding your store’s catalog, security, or data needs. Your store can grow at a faster rate when you don’t have to factor in new code for every single product or user.

# **What are some types of API used?**

There are 3 types of APIs used commonly today:

1. Open API
2. Partner API
3. Private API

Open APIs are publically available for anyone to use. BigCommerce, for example, uses roughly 25 different APIs, which is available for the public to use.

Partner APIs are designed by companies to offer API access to strategic business partners as an extra revenue channel for both parties. For example, Ticketmaster offers a Partner API to allow it’s clients the ability to reserve, buy, and retrieve ticket/event information.

Private APIs are not designed for public use and are designed for internal use. Let’s say you are traveling to a different city for a business meeting. You need to make a quick trip to the bank. You walk into “ABC Bank” and give the teller your account number. She quickly pulls up your account and you make a withdrawal. The teller was able to pull up your information by using ABC’s internal system, which uses an API to pull your account information and to update your new account balance.

# **Common Ecommerce APIs**

[Ecommerce APIs](https://www.bigcommerce.com/blog/?p=53002)are used in many ways. From displaying products on an online store to shipping them all over the world. APIs help owners manage their [online business](https://www.bigcommerce.com/blog/how-to-start-online-business/) and connect with customers fast and reliably.

Product information APIs

Product information APIs are on every ecommerce site, grabbing the information about your products and serving it to customers.

Site search APIs

The ability to site search isn’t automatic. Site searches need APIs to search through all your products containing a certain query and retrieve it for your user.

Payment APIs

If your online shop collects any form of electronic payments, you are using a payment API as the middleman between your shop and your processor.

Shipping APIs

Ever been to a site that asks you to put in your zip code to [calculate shipping](https://www.bigcommerce.com/ecommerce-answers/calculating-shipping-costs-ecommerce-business/)? That site is using an API with its shipping system or carrier to get you your best rate.

Currency conversion APIs

Buying shirts on a British site from a US IP used to be hard, now with currency conversion APIs, your favorite [international stores](https://www.bigcommerce.com/blog/international-ecommerce-expansion/) can convert currency in an instant. This API opens hundreds of thousands of online shops to international customers.

# The APIs Fueling Your BigCommerce Site

BigCommerce offers a range of APIs, below are some examples of our Catalog API, Login API, and Cart API that help power your BigCommerce store.

Catalog API.

Streamline catalog creation and editing to work smarter. Our catalog API uses fast automation to sync large catalogs in a matter of minutes, keeping inventory levels accurate without bogging down storefront performance. It takes just one API call to create a parent product with all variants and data. The BigCommerce catalog itself is the best in the industry, natively supporting multiple categories, complex products, physical and downloadable products, and gift certificates.

* *Source:*[*https://www.bigcommerce.com/product-catalog-api/*](https://www.bigcommerce.com/product-catalog-api/)

Login API.

Our Login API allows you to seamlessly connect your BigCommerce user accounts with your preferred login systems. Allow users to log in via your existing CMS software, single sign-on (SSO) systems or identity provider solutions. Using the proven SAML standard, you can support login via Active Directory, Google Apps, Okta, SailPoint, OneLogin and many others.

* *Source:*[*https://www.bigcommerce.com/login-api/*](https://www.bigcommerce.com/login-api/)

Cart API.

The BigCommerce Cart API allows you to tap directly into your store’s shopping cart. Access insightful cart data from your shoppers in order to power marketing initiatives and make better business decisions, with the ability to pass information into external CRM systems and analytics tools. You can also customize your cart experience with full read and write access.

* *Source:*[*https://www.bigcommerce.com/shopping-cart-api/*](https://www.bigcommerce.com/shopping-cart-api/)

# **Why use a custom API solution?**

Don’t get me wrong, already built integrations such as those available on the [Apps marketplace](https://www.bigcommerce.com/apps/) are wonderful tools. However, there are times when a custom integration is simply better for business. Custom-built integrations offer one added-bonus over pre-built apps, control. They are custom-tailored to your business’s specific needs.

Let’s say you are using an OMS (Order Management System/Software). While we have many integrations available on the Apps Marketplace, there isn’t one specific to your system. This is where a custom API solution makes sense. No dealing with the hassles and headaches that come with using a different system just because it is already integrated. You don’t have to do things such as retraining employees to use a new system, worrying about transferring important data across systems, etc. With the help of API, you can still use your current system, while having information automatically synced between BigCommerce and your OMS.

# **Conclusion**

API is a powerful tool that can help speed up your business operations, grow your brand’s reach, connect your shoppers to the products they want, and so much more. If you’re interested in finding out how API can help fuel your business. I’ve provided some more resources below.

The [API Developer Documentation](https://developer.bigcommerce.com/api-docs/) will show you guides on different API functions you can use.

Our [Apps Marketplace](https://www.bigcommerce.com/apps) can help you connect with things like Google Shopping, quickly back-up your important data with Rewind, or even help you build beautiful custom web pages with tools like Shogun.

Finally our[Partner Directory](https://www.bigcommerce.com/partners/) can connect you with various Certified BigCommerce Partners who are knowledgeable in the BigCommerce API and may be able to help with a custom integration that best suits your business.

| **Web Serviced** | **API** |
| --- | --- |
| All web services are APIs. | All APIs are not web services. |
| It supports XML. | Responses are formatted using Web API’s MediaTypeFormatter into XML, JSON, or any other given format. |
| You need a SOAP protocol to send or receive and data over the network. Therefore it does not have light-weight architecture. | API has a light-weight architecture. |
| It can be used by any client who understands XML. | It can be used by a client who understands JSON or XML. |
| Web service uses three styles: REST, SOAP, and XML-RPC for communication. | API can be used for any style of communication. |
| It provides supports only for the HTTP protocol. | It provides support for the HTTP/s protocol: URL Request/Response Headers, etc. |