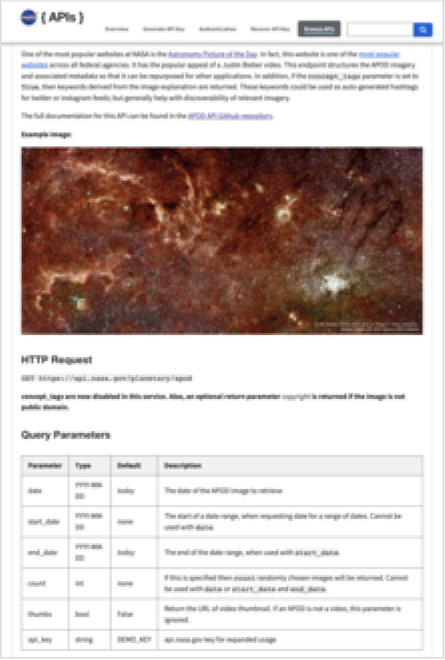
Wikipedia
The Free Encyclopedia

Web API

A **web API** is an application programming interface (API) for either a web server or a web browser. As a web development concept, it can be related to a web application's client side (including any web frameworks being used). A server-side web API consists of one or more publicly exposed **endpoints** to a defined request–response message system, typically expressed in JSON or XML by means of an HTTP-based web server. A server API (SAPI) is not considered a server-side web API, unless it is publicly accessible by a remote web application.



One of the most popular websites at NASA is the Astronomy Picture of the Day. In fact, this website is one of the most popular websites across all federal agencies. It has the popular appeal of a Justin Bieber video. This endpoint structures the APOD imagery and associated metadata so that it can be repurposed for other applications. In addition, if the concept_tags parameter is set to True, then keywords derived from the image explanation are returned. These keywords could be used as auto-generated hashtags for twitter or instagram feeds; but generally help with discoverability of relevant imagery.

The full documentation for this API can be found in the APOD API Github repository.

Example image:

HTTP Request

`GET https://api.nasa.gov/planetary/apod`

concept_tags are now disabled in this service. Also, an optional return parameter copyright **is returned if the image is not public domain.**

Query Parameters

Parameter	Type	Default	Description
date	YYYY-MM-DD	today	The date of the APOD image to retrieve
start_date	YYYY-MM-DD	none	The start of a date range, when requesting date for a range of dates. Cannot be used with date.
end_date	YYYY-MM-DD	today	The end of the date range, when used with start_date.
count	int	none	If this is specified then count randomly chosen images will be returned. Cannot be used with date or start_date and end_date.
thumbs	bool	False	Return the URL of video thumbnail. If an APOD is not a video, this parameter is ignored.
api_key	string	DEMO_KEY	api.nasa.gov key for expanded usage

Screenshot of web API documentation written by NASA

Client side

A client-side web API is a programmatic interface to extend functionality within a web browser or other HTTP client. Originally these were most commonly in the form of native plug-in browser extensions however most newer ones target standardized JavaScript bindings.

The Mozilla Foundation created their WebAPI specification which is designed to help replace native mobile applications with HTML5 applications.[1][2]

Google created their Native Client architecture which is designed to help replace insecure native plug-ins with secure native sandboxed extensions and applications. They have also made this portable by employing a modified LLVM AOT compiler.

Server side

A server-side web API consists of one or more publicly exposed *endpoints* to a defined request–response message system, typically expressed in JSON or XML. The web API is exposed most commonly by means of an HTTP-based web server.

Mashups are web applications which combine the use of multiple server-side web APIs.[3][4][5] *Webhooks* are server-side web APIs that take input as a Uniform Resource Identifier (URI) that is designed to be used like a remote named pipe or a type of callback such that the server acts as a client to dereference the provided URI and trigger an event on another server which handles this event thus providing a type of peer-to-peer IPC.

Endpoints

Endpoints are important aspects of interacting with server-side web APIs, as they specify where resources lie that can be accessed by third party software. Usually the access is via a URI to which HTTP requests are posted, and from which the response is thus expected. Web APIs may be public or private, the latter of which requires an *access token*.[6]

Endpoints need to be static, otherwise the correct functioning of software that interacts with them cannot be guaranteed. If the location of a resource changes (and with it the endpoint) then previously written software will break, as the required resource can no longer be found at the same place. As API providers still want to update their web APIs, many have introduced a versioning system in the URI that points to an endpoint.

Resources versus services

Web 2.0 Web APIs often use machine-based interactions such as REST and SOAP. RESTful web APIs use HTTP methods to access resources via URL-encoded parameters, and use JSON or XML to transmit data. By contrast, SOAP protocols are standardized by the W3C and mandate the use of XML as the payload format, typically over HTTP. Furthermore, SOAP-based Web APIs use XML validation to ensure structural message integrity, by leveraging the XML schemas provisioned with WSDL documents. A WSDL document accurately defines the XML messages and transport bindings of a Web service.

Documentation

Server-side web APIs are interfaces for the outside world to interact with the business logic. For many companies this internal business logic and the intellectual property associated with it are what distinguishes them from other companies, and potentially what gives them a competitive edge. They do not want this information to be exposed. However, in order to provide a web API of high quality, there *needs* to be a sufficient level of documentation. One API provider that not only provides documentation, but also links to it in its error messages is Twilio.[7]

However, there are now directories of popular documented server-side web APIs.[8]

Growth and impact

The number of available web APIs has grown consistently over the past years, as businesses realize the growth opportunities associated with running an open platform, that any developer can interact with. ProgrammableWeb tracks over 24000 Web APIs that were available in 2022, up from 105 in 2005.

Web APIs have become ubiquitous. There are few major software applications/services that do not offer some form of web API. One of the most common forms of interacting with these web APIs is via embedding external resources, such as tweets, Facebook comments, YouTube videos, etc. In fact there are very successful companies, such as Disqus, whose main service is to provide embeddable tools, such as a feature-rich comment system.[9] Any website of the TOP 100 Alexa Internet ranked websites uses APIs and/or provides its own APIs, which is a very distinct indicator for the prodigious scale and impact of web APIs as a whole.[10]

As the number of available web APIs has grown, open source tools have been developed to provide more sophisticated search and discovery. APIs.json provides a machine-readable description of an API and its operations, and the related project APIs.io offers a searchable public listing of APIs based on the APIs.json metadata format.[11][12]

Business

Commercial

Many companies and organizations rely heavily on their Web API infrastructure to serve their core business clients. In 2014 Netflix received around 5 billion API requests, most of them within their private API.[13]

Governmental

Many governments collect a lot of data, and some governments are now opening up access to this data. The interfaces through which this data is typically made accessible are web APIs. Web APIs allow for data, such as "budget, public works, crime, legal, and other agency data"[14] to be accessed by any developer in a convenient manner.

Example

An example of a popular web API is the Astronomy Picture of the Day API operated by the American space agency NASA. It is a server-side API used to retrieve photographs of space or other images of interest to astronomers, and metadata about the images.

According to the API documentation,[15] the API has one endpoint:

```
https://api.nasa.gov/planetary/apod
```

The documentation states that this endpoint accepts GET requests. It requires one piece of information from the user, an API key, and accepts several other optional pieces of information. Such pieces of information are known as *parameters*. The parameters for this API are written in a format known as a query string, which is separated by a question mark character (?) from the endpoint. An ampersand (&) separates the parameters in the query string from each other. Together, the endpoint and the query string form a URL that determines how the API will respond. This URL is also known as a *query* or an *API call*.

In the below example, two parameters are transmitted (or *passed*) to the API via the query string. The first is the required API key and the second is an optional parameter — the date of the photograph requested.

```
https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY&date=1996-12-03
```

Visiting the above URL in a web browser will initiate a GET request, calling the API and showing the user a result, known as a *return value* or as a *return*. This API returns JSON, a type of data format intended to be understood by computers, but which is somewhat easy for a human to read as well. In this case, the JSON contains information about a photograph of a white dwarf star:

```
{
  "date":"1996-12-03",
  "explanation":"Like a butterfly,\r a white dwarf star begins its life\r by casting off a
cocoon that enclosed its former self. In this\r analogy, however, the Sun would be\r a
caterpillar\r and the ejected shell of gas would become the prettiest of all!\r The above
cocoon, the planetary nebula\r designated NGC 2440, contains one of the hottest white dwarf
stars known.\r The white dwarf can be seen as the bright dot near the photo's\r center. Our
Sun will eventually become a \"white dwarf butterfly\",\r but not for another 5 billion
years. The above false color image recently entered the public domain\r and was post-
processed by F. Hamilton.\r",
  "hdurl":"https://apod.nasa.gov/apod/image/9612/ngc2440_hst2_big.jpg",
  "media_type":"image",
  "service_version":"v1",
  "title":"Cocoon of a New White Dwarf\r\nCredit:",
  "url":"https://apod.nasa.gov/apod/image/9612/ngc2440_hst2.jpg"
}
```

The above API return has been reformatted so that names of JSON data items, known as *keys*, appear at the start of each line. The last of these keys, named `url`, indicates a URL which points to a photograph:

```
https://apod.nasa.gov/apod/image/9612/ngc2440_hst2.jpg
```

Following the above URL, a web browser user would see this photo:

Although this API can be called by an end user with a web browser (as in this example) it is intended to be called automatically by software or by computer programmers while writing software. JSON is intended to be parsed by a computer program, which would extract the URL of the photograph and the other metadata. The resulting photo could be embedded in a website, automatically sent via text message, or used for any other purpose envisioned by a software developer.

See also

- Application programming interface
- Dynamic web page
- Overview of RESTful API Description Languages
- REST

- GraphQL
- Web service

References

1. "Mozilla's WebAPI Wants to Replace Native Apps With HTML5 | Webmonkey | Wired.com" (https://web.archive.org/web/20131205024538/http://www.webmonkey.com/2011/08/mozillas-webapi-wants-to-replace-native-apps-with-html5/). 2013-12-05. Archived from the original (http://www.webmonkey.com/2011/08/mozillas-webapi-wants-to-replace-native-apps-with-html5/) on 2013-12-05. Retrieved 2022-01-13.
2. Mozilla WebAPI Wiki (https://developer.mozilla.org/en-US/docs/Web/API)
3. "What is mash-up? - Definition from WhatIs.com" (http://whatis.techtarget.com/definition/mash-up). *WhatIs.com*. Archived (https://web.archive.org/web/20150424163155/http://whatis.techtarget.com/definition/mash-up) from the original on Apr 24, 2015. Retrieved 2015-11-04.
4. "Mashup Dashboard" (https://web.archive.org/web/20210507015502/http://www.programmableweb.com/mashups). ProgrammableWeb.com. 2009. Archived from the original (http://www.programmableweb.com/mashups) on May 7, 2021. Retrieved May 29, 2021.
5. Maximilien, E. Michael; Ranabahu, Ajith; Gomadam, Karthik (Sep–Oct 2008). "An Online Platform for Web APIs and Service Mashups". *IEEE Internet Computing*. **12** (5): 32–43. doi:10.1109/MIC.2008.92 (https://doi.org/10.1109%2FMIC.2008.92). S2CID 14949046 (https://api.semanticscholar.org/CorpusID:14949046).
6. "Changes coming in Version 1.1 of the Twitter API" (https://blog.twitter.com/developer/en_us/a/2012/changes-coming-to-twitter-api.html).
7. Mulloy, Brian. *Web API Design - Crafting Interfaces that Developers Love* (https://pages.apigee.com/rs/apigee/images/api-design-ebook-2012-03.pdf) (PDF). apigee. p. 11.
8. "API Directory" (http://www.programmableweb.com/category/all/apis?order=field_popularity). *ProgrammableWeb*. Retrieved 2015-11-03.
9. "Disqus – The Web's Community of Communities" (https://disqus.com/). *Disqus*. Retrieved 2015-11-04.
10. "Alexa Top 500 Global Sites" (http://www.alexa.com/topsites). *www.alexa.com*. Retrieved 2015-11-04.
11. "APIs.json" (http://apisjson.org/). *apisjson.org*. Retrieved 2016-03-14.
12. "APIs.io - the API search engine" (http://apis.io/). *apis.io*. Retrieved 2016-03-14.So make it clearly understandable & share your knowledge.
13. "Top 10 Lessons Learned from the Netflix API - OSCON 2014, Slide 73" (https://www.slideshare.net/danieljacobson/top-10-lessons-learned-from-the-netflix-api-oscon-2014). 2014-07-24. {{cite journal}}: Cite journal requires |journal= (help)
14. "Tech Trends 2015, API economy" (http://dupress.com/articles/tech-trends-2015-what-is-api-economy/). *Deloitte University Press*. Retrieved 2015-11-03.
15. "Astronomy Picture of the Day" (https://api.nasa.gov/). *NASA APIs*.

Further reading

- Jacobson, Daniel; Woods, Dan; Brail, Greg (November 2011). *APIs a strategy guide*. Sebastopol, Calif: O'Reilly Media. ISBN 978-1-4493-0892-6.
- Mulloy, Brian. *Web API Design - Crafting Interfaces that Developers Love* (https://pages.apigee.com/rs/351-WXY-166/images/Web-design-the-missing-link-ebook-2016-11.pdf) (PDF). Apigee.
- "Web APIs" (https://developer.chrome.com/extensions/api_other). *Google Chrome*.

- "Web API reference - Web technology reference" (https://developer.mozilla.org/en-US/docs/Web/Reference/API). *Mozilla Developer Network*.
- "Web API Demonstration" (http://sourcecodehub.com/article/428/api-demonstration-aspnet). *Web API Demonstration*.

-