

⚡ Current Skill Servers and API

Introduction

API is the acronym for Application Programming Interface. It is a software intermediary that allows two applications to speak to each other.

An **API** is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange.

APIs serve numerous purposes. Generally, they can simplify and speed up software development.

Whenever you use an application like Facebook, send an instant message, or check the weather on your phone, you're using an API.

What is an API?



What Are the Advantages of APIs?

"There is no information exchange without an API. There is no digital transformation without an API. The Open API is the real revolution!" - Bertrand Lafforgue, > Microsoft Commercial Data Platform Director.

- **Automation:** With APIs, computers rather than people can manage the work. Using APIs, agencies can update workflows to make them faster and more productive.
- **Application:** APIs can access application components. The delivery of services and information is more flexible.

- **More scope:** Through the use of an API an application layer can be created which can also be used to distribute information and services to new users. These APIs can be customized to create personalized user experiences.
- **Unlimited access to Data:** An API allows all public information that is generated by the government for general use to be accessible to all citizens without any exclusivity.
- **Efficiency:** When access is provided to an API, the generated content can be published automatically and is available for each channel. It makes it easier to share and distribute.
- **Integration:** APIs make it easier to embed content from any site or application. This ensures smoother information flow and an integrated user experience.
- **Personalization:** Thanks to APIs any user or company can personalize the content and services they use the most.
- **Adaptation:** Needs evolve and APIs allow you to anticipate changes. When working with this technology data transfer is easily supported and the information is examined more closely.

Basically, APIs are developer-friendly, easily accessible, and incredibly flexible.

REST API

REST (Representational State Transfer) is a very popular web API architecture. REST API is an API that must have certain architectural constraints or principles including:

- **Client-server architecture:** The interface is separated from the backend and data storage. This allows for flexibility and for different components to evolve independently.
- **Statelessness:** No client context is stored on the server between requests.
- **Cacheability:** Clients can cache responses, so a REST API response must explicitly state whether it can be cached or not.
- **Layered system:** The API will work whether it is communicating directly with a server or through an intermediary, such as a load balancer.

The REST architecture uses the original instructions of the HTTP protocol.



Rule # 1: The URL must operate as a resource identifier.

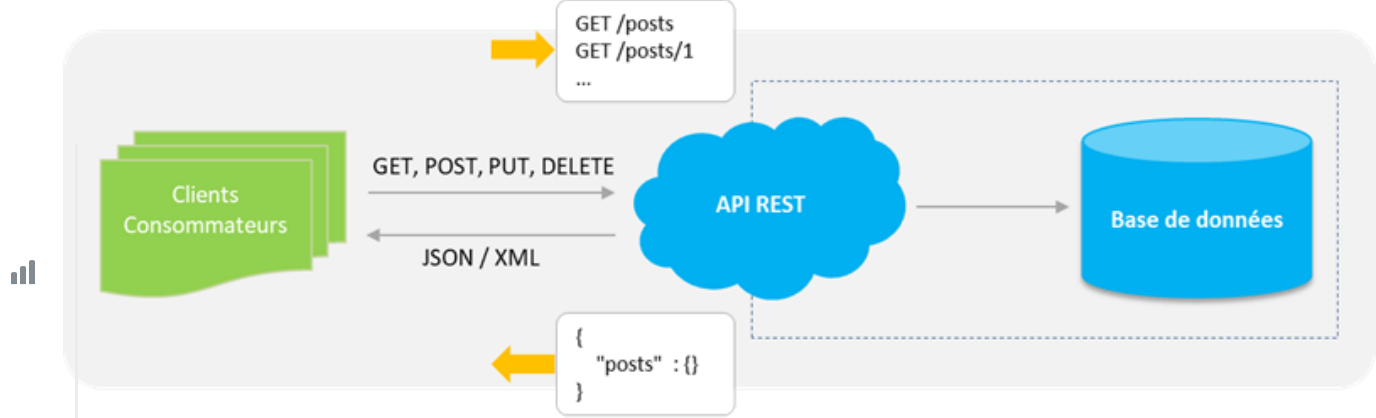
Rule # 2: HTTP verbs must function as the identifiers of operations.

Rule # 3: HTTP responses must operate as representations of resources.

Rule # 4: Links are like a relationship between resources.



Rule # 5: A parameter, like an authentication token, must exist.



HTTP Methods

HTTP methods define a set of query methods that indicate the action that we want to perform on the indicated resource. Although they have names, these methods are often called HTTP verbs. Each of them implements a different semantic, but some common functionalities can be shared by different methods.

GET: The GET method requests a representation of the indicated resource. GET requests should only be used to retrieve data.

HEAD: The HEAD method requests a response identical to a GET request but the body of the response is omitted (we only have the header).

POST: The POST method is used to send an entity to the indicated resource. This usually results in a change of state or causes side effects on the server.

PUT: The PUT method replaces all current representations of the resource targeted by the content of the request.

DELETE: The DELETE method deletes the specified resource.

CONNECT: The CONNECT method builds a tunnel to the server identified by the target resource.

OPTIONS: The OPTIONS method is used to describe the communication options with the targeted resource.

TRACE: The TRACE method performs a round trip test message by following the path of the targeted resource.

PATCH: The PATCH method is used to apply partial modifications to a resource.