

What are REST and Streaming APIs

Edris Safari

Bellevue University, Nebraska U.S.A.

### Abstract

The concept of API has gradually come to reality since the advent of computer programming. Application programming interfaces are “providers” of services that an application provides. Both REST and streaming APIs leverage from this concept. We will describe each and compare their functions in the modern age of world wide web and consequently data science.

*Keywords:*

API – Application Programming Interface

REST – **R**epresentational **S**tate **T**ransfer

## What are REST and Streaming APIs

### Introduction

In the early days of computing, APIs came in the form of DLLs (Dynamically Loaded Libraries), OCX(OLE(Object linking and embedding) Control Extension), ActiveX, libraries and to some degree SDKs(Software Development Kit). These APIs allowed programmers to use the functions they provide by “calling” functions that were built to perform certain tasks such as resizing a window, or changing the color of text, etc. They reduce or eliminate recoding things that can be coded once.

In the modern age of world wide web, API's have become a mechanism for applications to “communicate” with one another in local (i.e. Intranet) or distributed (i.e. Cloud) environments. Powered by Object Oriented Programming techniques and distributed computing, APIs have now taken the form of web services. Just like the traditional API's, web services (and objects) have methods and attributes. If someone wants to use the web service (i.e. return a list of houses in zip code 78654), she/he would invoke the appropriate web service method with the appropriate attributes.

Whether old or new API's, they all must have certain set of rules that must be known by the consumers of the API. We can think of these rules as sort of agreement between the API and the use of the API. The start button is an API because no matter where we use it, pushing it would start something. It is agreed by all users of the start button, that this is the effect of using the start API.

REST APIs are not really a set of APIs that we can invoke via web service or any other mechanism. REST API is framework wherein components of a networked environment can exchange data. The REST architectural style specifies a uniform interface that facilitates

properties such as performance, scalability, and modifiability. These properties enable services on the web to exchange data seamlessly (and reliably).

In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource identifiers (URIs), typically and most commonly linked on the web. It is a client/server architecture and is designed to use a stateless communication protocol such as HTTP. Clients and servers exchange the representations of resources by using a standard interface protocol such as JSON.

REST provides a standard way for a web service provider to expose its resources on the web. Similarly, the client accesses those services via the HTTP methods GET, PUT, DELETE, and POST. The resources can be considered as objects that represent the resource. Since the objects are represented in markup languages, manipulations of them becomes standard. For example, the python library for JSON and Beautiful Soup can be used to parse a reply to GET. In the REST framework, the rules to qualify and API as REST API , or REST Web Service or RESTFul APIs, the following HTTP requests must be supported.:

GET – Used to retrieve information from the endpoint

POST – Used to create a new entity on the endpoint or to update an existing entity

PUT – Used to create a new entity or to update an existing entity. The most significant difference between POST and PUT is that PUT is idempotent. This means repeated calls of these methods has the same effect on the server side, but not necessarily on the client side.

DELETE –Used to request that a resource be removed. DELETE is also idempotent except the reply to 1st DELETE is different than subsequent ones because the object does not exist in subsequent times so the client will get a "No Content" or "Not found" error.

PATCH – Used to request a partial update to a resource.

Conceptualized by Roy Fielding, REST is the underlying architecture of the web because it uses the HTTP protocol. APIs developed using the REST framework can provide different services. In a push/pull environment, the APIs communicate with their clients (or other server for that matter) in a request/reply fashion. In other cases, it is a polling mechanism where clients (or servers for that matter) would constantly poll for data. In other cases, live streaming of data is provided by the API. In all cases, the API's rules must be followed regarding security, format of the messaging (XML, HTML, JSON, etc.)

### **Conclusion**

REST APIs are APIs that follow the HTTP protocol. Additionally, they provide for security mechanism and exchange of data in various formats. From this framework, we can build APIs that can provide services specific to the use case. Live streaming APIs aren't just say streaming video, but rather continuous feed of data. For example, a streaming API could notify a client of a customer's spending habit. The client can detect a fraudulent charge as soon as one comes in that doesn't fit the norm.

### References

1. [APIs | REST | REST APIs Demystified](#)
2. [Python Requests Tutorial: Request Web Pages, Download Images, POST Data, Read JSON, and More](#)
3. Sarkar, Dr. Tirthajyoti. Data Wrangling with Python: Creating actionable data from raw sources . Packt Publishing. Kindle Edition.
4. Going Real-Time: How to Build a Streaming API -  
<https://www.youtube.com/watch?v=4Awwux4hKYE>
5. RESTful API (REST API) -  
<https://searchapparchitecture.techtarget.com/definition/RESTful-API>