**Content**

[1. Nadpis 1 5](#_Toc3283217)

1. Batch, Service, API, Web Service, Rest, Soap, XML, HTTP and friends
   1. Batch vs As a Service

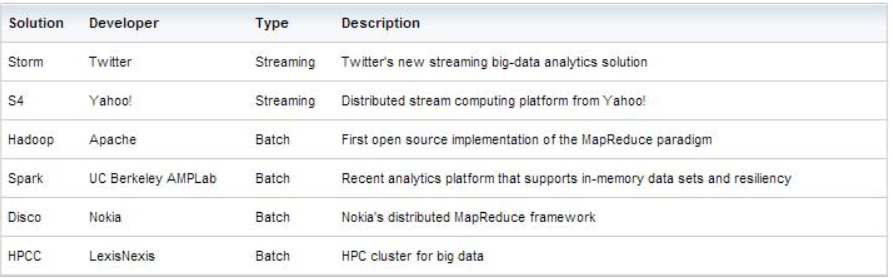
You'll use a batch job if you want to run the functionality on schedule. You'll use a service if you want it to run on a request from another application.

The main difference between batch jobs and real-time jobs is that a real-time job (when configured as a real-time service) is running constantly. It waits for input messages, executes, and returns a result - which can contain many elements. An example might be a validation engine, it would accept many (potentially parallel) input messages and reply (hopefully) in a very short time. It can support a high level of throughput (if designed correctly).

When batch jobs are executed, they start processes to perform transformation and cannot return a complex reply back to the initiator.

Calling both types via the web service are valid and realistic scenarios, a real-time job would be used for fast processing of small input messages and consumable replies, a batch job could be triggered by an enterprise scheduler once a week to export a large volume of data to a 3rd party.

List of batch and real-time processing tools:



* + 1. Batch data processing

Batch data processing is an efficient way of processing **high volumes of data** is where a group of transactions is **collected over a period of time.** Data is collected, entered, processed and then the batch results are produced (**Hadoop** is focused on batch data processing). Batch processing requires separate programs for input, process and output. An example is **payroll and billing systems.**

* + 1. Real-Time jobs

In contrast, real time data processing involves a continual input, process and output of data. Data must be processed in a small time period (or near real time). Radar systems, customer services and bank ATMs are examples.

* 1. What is web service?

A web service is a resource that can be accessed on a network. Usually this via the HTTP protocol, comforming to the Simple Object Access Protocol (SOAP) or Representational state transfer (REST). Providing processing and/or storage functionality as a web service, makes sense for the purposes of sharing resources, security, distributing load, and other reasons. It is often on a separate machine, but this is not a requirement.

A web framework, in comparison, is a generic architecture template which assists the development of not just web services, but browser-accessible web applications as well. Often a web framework would provide some sort of model-view-controller architecture, where all the "business logic" is handled by the controller component, the database access is handled by the model component, and the interface to or output representation of the data is provided by the view component. A web framework may, as is typically with a REST API, automatically render an output representation of the data (via a JSON library, for instance), negating the need for a view component.

* + 1. API vs Web Service

API and Web service serve as a means of communication. The only difference is that a Web service facilitates interaction between two machines over a network. An API acts as an interface between two different applications so that they can communicate with each other. An API is a method by which the third-party vendors can write programs that interface easily with other programs. A Web service is designed to have an interface that is depicted in a machine-processable format usually specified in Web Service Description Language (WSDL). Typically, “HTTP” is the most commonly used protocol for communication. Web service also uses SOAP, REST, and XML-RPC as a means of communication. API may use any means of communication to initiate interaction between applications. For example, the system calls are invoked using interrupts by the Linux kernel API.

**Summary:**

1. All Web services are APIs but all APIs are not Web services.
2. Web services might not perform all the operations that an API would perform.
3. A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication whereas API may use any style for communication.
4. A Web service always needs a network for its operation whereas an API doesn’t need a network for its operation.
5. An API facilitates interfacing directly with an application whereas a Web service interacts with two machines over a network.
6. Web service is like advanced Urls and API is Programmed Interface.
7. API contains classes and Interfaces just like a program.
8. A web service is a form of API (Application Programming Interface).
9. An API is used by a computer programmer to establish a link between software applications. This interface can take several forms, a web service is just one of these.
10. There are several types of web service. SOAP (Simple Object Access Protocol) is one of the most common. The API takes the form of a service description (WSDL) which is used to automatically generate the program code which makes the connection.