

# Salesforce Streaming Connector

## What is streaming

Streaming is a method used to transfer data over a network as a continuous flow. One of its main benefits is that it allows the data recipients to use it in real time, without having to wait until the complete data set has been uploaded on their computers.

## What is the Salesforce Streaming API

Salesforce has a set of data APIs that can be used to manipulate the App's data. Among them, the Streaming API is used to set notifications triggered by data changes. The notifications are sent from a server to a client.

The Salesforce Streaming API follows the Publish – Subscribe model. In it, users subscribe to available channels that broadcast specific data changes. These data changes are defined by SOQL queries. For example, a channel can notify its users about changes in Account's Emails.

The main advantage of this model is that it eliminates pooling and therefore, it is useful for communicating very frequent changes - practically producing real time integration.

Additionally, the API's use of long polling or Comet Programming technique, reduces the number of requests that return no-data.

## Main Characteristics of the Salesforce Streaming API

Salesforce API has many important characteristics that define its behavior. Among them:

1. **Communication type:** the API is based on asynchronous communication.
2. **Bayeux Protocol:** the API uses this protocol to transmit asynchronous messages.
3. **CometD:** the API uses this scalable event routing bus for message transmission. It is based on Ajax communication and the Bayeux protocol. The method opens a single persistent connection from the client to the server. Events are then incrementally handled and interpreted on the Client's side, every time a new event arrives.
4. **Long polling or Comet programming:** this method emulates an information push from server to client. As it waits until there is some data to be sent, it doesn't send no-data notifications.

## The DBSync Salesforce Streaming Connector

It is within the context of the Salesforce Streaming API that, DBSync developed a connector that can be used specifically for real time integration. The connector is known as Salesforce Streaming Connector.

There are two important points to consider when using the Salesforce Streaming Connector. First, the process should run as continuously as possible, without much interruption. Second, how to schedule the process. We will explain the use of this connector with an example.

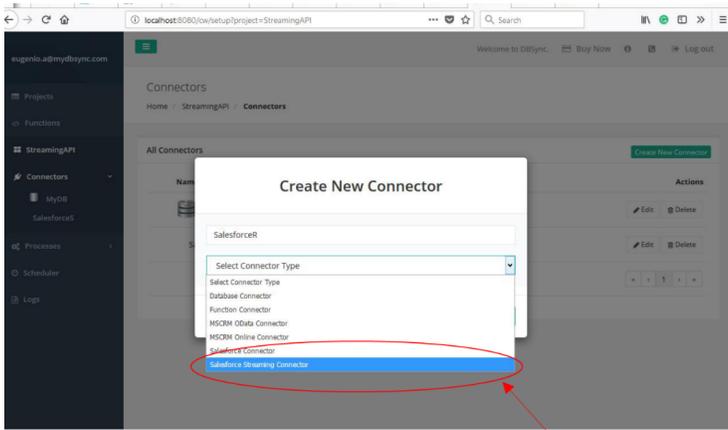
In this example we will connect Salesforce table Accounts to a MySQL table Accounts. The steps are:

1. Create a new project.
2. Create a Salesforce Streaming Connector and a Database Connector.
3. Create a new process.
4. Create a new workflow.
5. Create a new trigger, rule and map.
6. Create a schedule.
7. Run the process.

As the steps are similar to any DBSync process creation, we will concentrate on the steps with specific differences, namely steps 2, 4 and 5.

## Create a Salesforce Streaming Connector

In order to create a Salesforce Streaming connector, simply press the button *Create New Connector*, and select the type *Salesforce Streaming Connector*.



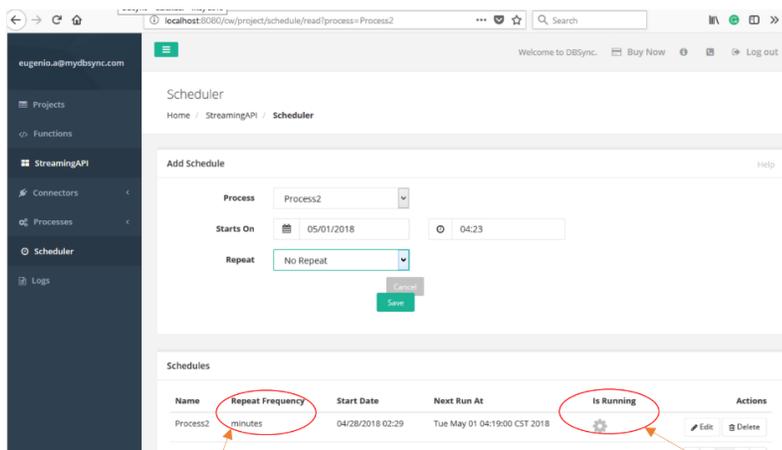
Select a *Salesforce Streaming Connector*

Figure 1

## Create a Schedule

As we mentioned earlier, we want to create a process that runs as continuously as possible. However, DBSync cannot detect if the process has stopped for any reason; and thus, it cannot restart it automatically.

In order to solve this problem, we need to set up the schedule at every one minute interval. The cron expression for this is "0/1 0/1 1/1 ?\*". Alternatively, we can simply select minutes and specify "1".



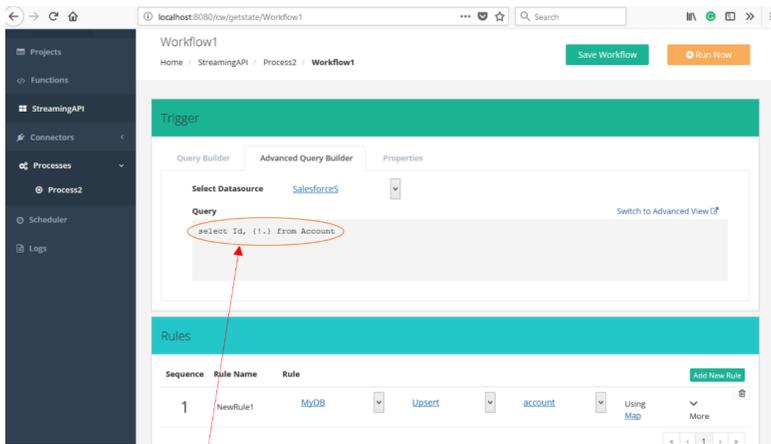
Scheduler is set up with a repeat frequency of one minute

The process remains running

Figure 2

## Create a Trigger

As we are transferring data from Salesforce to MySQL, we need to create a trigger that contains a query. This query must include the ID of the Salesforce table - in our case Accounts. The query must not include joins or aggregate functions. Besides, the WHERE clause must not include lookup fields.



Query must include Id

Figure 3

## Create a New Rule

When we create a new rule, we must make sure to map the ID from Salesforce to the MySQL table's ID.

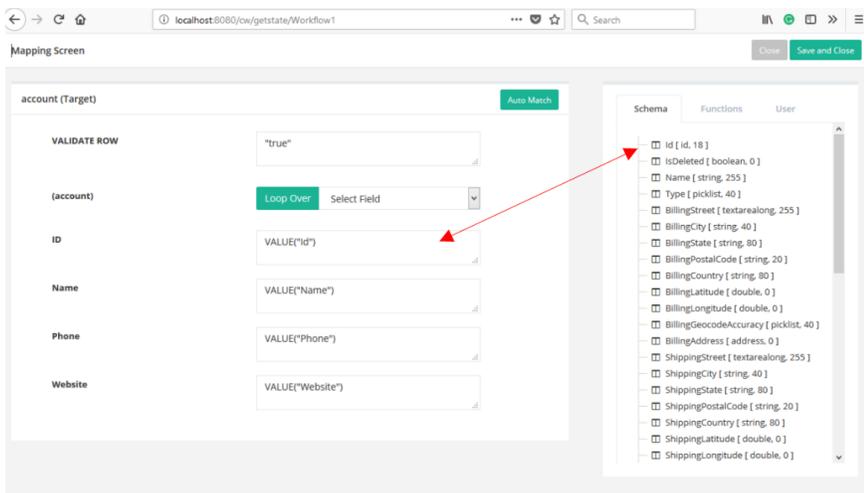
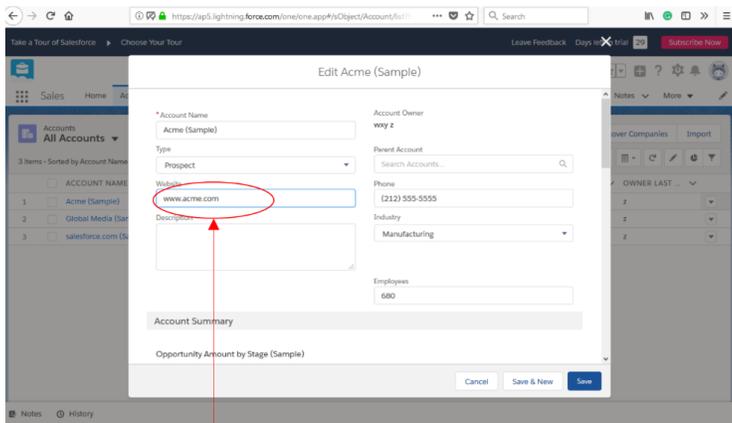


Figure 4

## Seeing the Results

The advantages of the Salesforce Streaming Connector can be easily seen by changing the value of, for example, the website of an account: the change is almost immediately reflected in our MySQL database.



We add a website for the ACME account

Figure 5

The change appears almost immediately on the database

ID	Name	Phone	Website
0017F00000gTWqQAW	A. Dahum		adahum@adahum.com
0017F00000gTWqQAW	Acme (Sample)	(212) 555-5555	www.acme.com
	Adventure Works		Adrian@adventure-works.com
	Alpine Ski House		Cathan@alpineskihouse.com
	Best of Sales		
	Blue Yonder Airlines		brian@blueyonderairlines.com
	City Power & Light		walter@cpandl.com
	Clients		
	Coho Winery		Cat@cohowinery.com
	Consolidated Messenger		Forrest@consolidatedmessenger.com
	Fabrikam, Inc.		evacorets@fabrikam.com
	Fourth Coffee		gabriele@fourthcoffee.com
0017F00000gTWqQAW	Global Media (Sample)	(905) 555-1212	
	Graphic Design Institute		george@graphicdesigninstitute.com
	Humongous Insurance		TSaleh@humongousinsurance.com
	Litware		andrew@litware.com
	Lucerne Publishing		psteiner@lucernepublishing.com
	Margie's Travel		sueburk@margiestravel.com
	Microsoft		contactus@microsoft.com
	Northwind Traders		alexw@northwindtraders.com
	Proseware, Inc.		ymckay@proseware.com
0017F00000gTWqQAW	salesforce.com (Sample)	(415) 901-7000	http://www.salesforce.com
	School of Fine Art		info@schoolofart.net

Figure 6

## Conclusion

DBSync's Salesforce Streaming Connector allows us to link Salesforce to another application in almost real time.

However, there are few aspects to consider - namely, the setting up of the schedule with a repeating frequency of one minute; the limitations on the trigger's query; and, the mapping of IDs.