

QuickBooks to Database Integration

Have you ever thought about integrating QuickBooks to a Database of your choice? The benefits are legion - from backing up your data in QuickBooks, to copying selected data for use in Analytics or other business tools.

In this article, you will learn how to create the integration with DBSync. The tool is user-friendly - making integration a breeze.

The results are fast. By defining workflows, you can quickly create different integration paths to meet your business needs. Workflows can be run manually or automatically via a scheduler.

The integration process involves nine steps. The steps are explained below by providing an example.

Preliminary step

In order to show how to integrate QuickBooks into a Database, we will use an example of how to transfer Accounts data from QuickBooks to a MySQL Database.

In order to repeat this example, you first need to create a MySQL Database named "qbtodb". This database must contain a table named "Accounts". The table must contain two fields: "AccountName" and "AccountAmount".

Although the example shows how to transfer records from QuickBooks to a Database, the process could be defined inversely. That is, you could do a transfer from a Database to QuickBooks in a similar manner. That is, by just changing the source and destination data.

So, let's start with our integration process.

Step 1: Create a Project

In order to integrate QuickBooks to a Database, the first thing that you need to do is to create a Project that will contain all the necessary information.

For this, click on the *Create New Project* button and enter a name of your choice. Remember two important points:

1. Project names cannot include blank spaces.
2. It is a good practice to use names related to what your project will do.

Once you have entered a name, click on the *Save* button. Your project will appear in the lists of available projects.

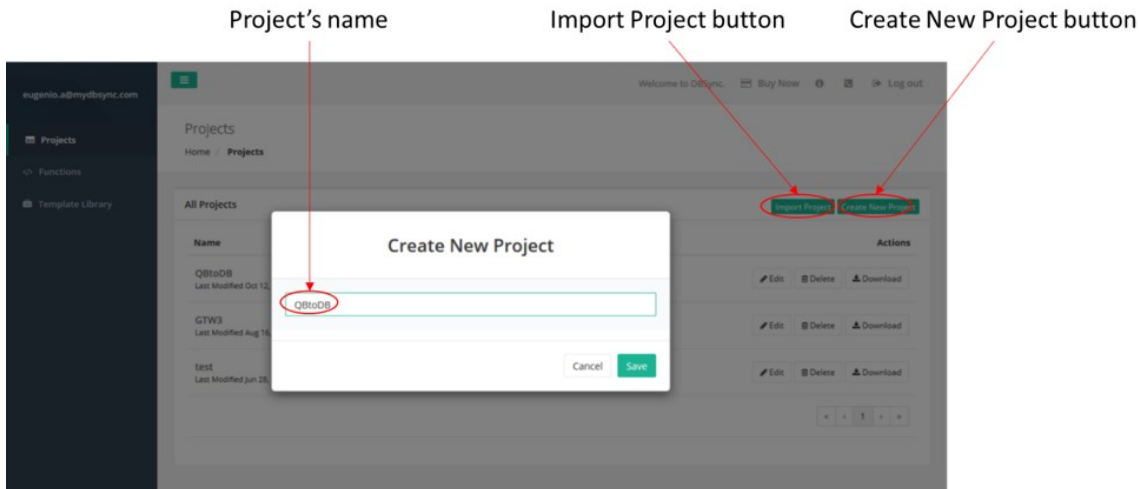


Figure 1: Create a new project.

Figure 2: List of projects.

Step 2: Create the QuickBooks connector

Your second step is to create a Connector that will allow you to send and retrieve information from your QuickBooks app.

In order to do so, you need to click on the *Create New Connector* button. The system will ask for a Connector's name and type. Enter any name.

Just remember that a Connector's name cannot include any space between letters. For the Connector's type, select *QuickBooks Online Rest Connector*.

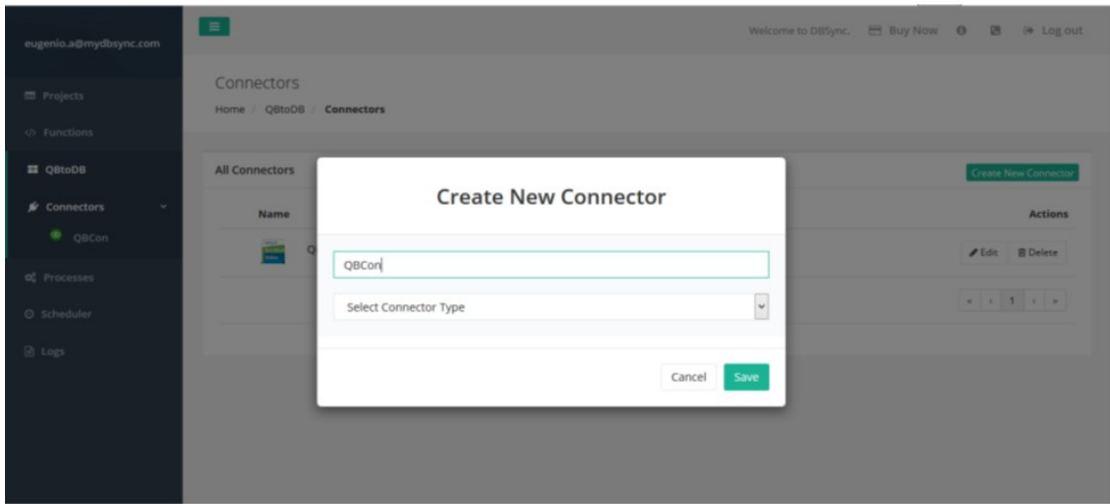


Figure 3: Connector's name

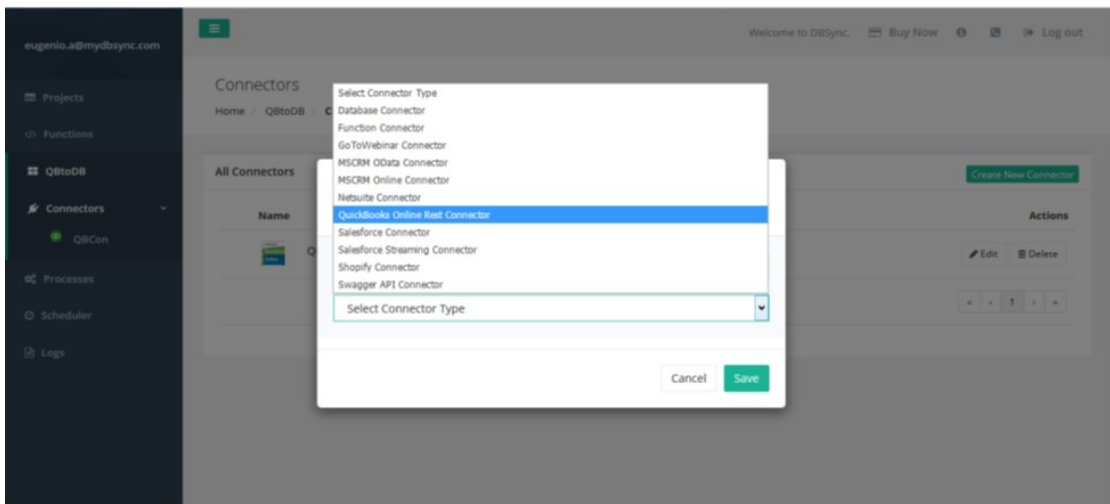


Figure 4: Connector's type menu

Now the system will ask for your QuickBook credentials. Enter them and click on the *Connect* button. The Connector's name will appear on the left-hand side menu. And, the setting parameters will be on the right-hand screen.

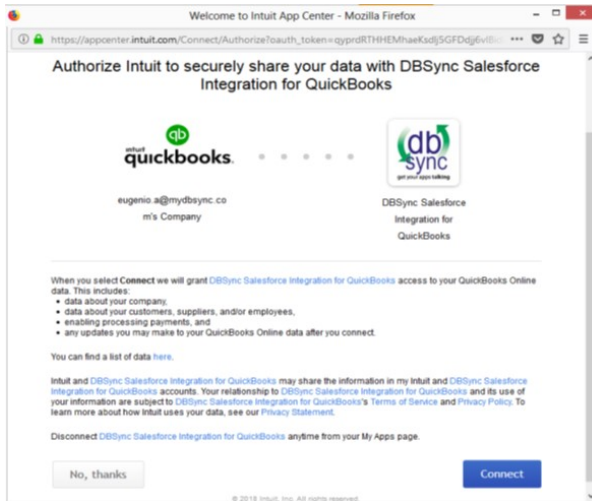


Figure 5: Quickbook's credentials.

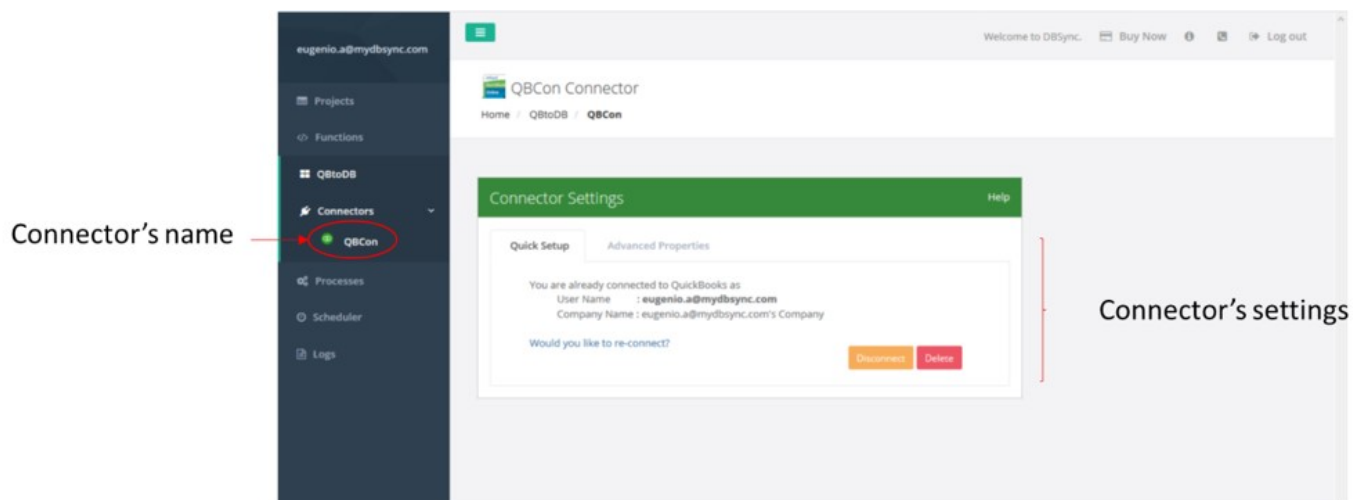


Figure 6: Connector's details.

Step 3: Create the Database Connector

Your next step is to create a Connector to your Database. In order to do this, click on the *Create New Connector* button again.

A screen asking for a Connector's name and type will appear. Enter a name of your choice and select *Database Connector* as type. Then, click on the *Save* button.

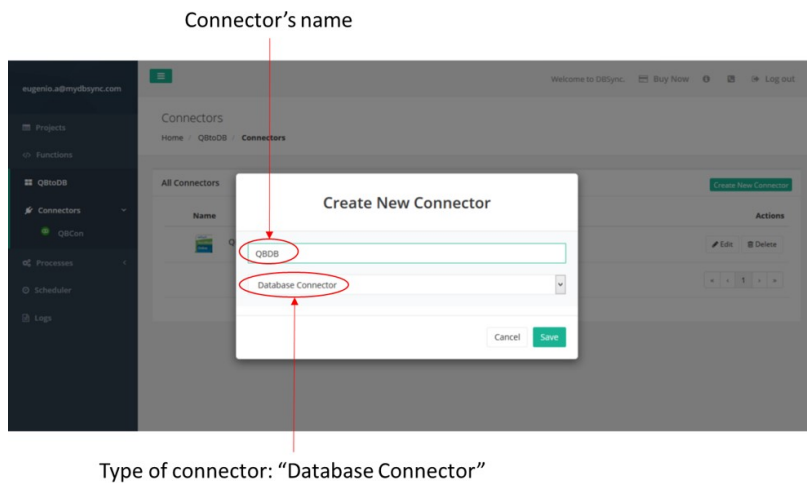


Figure 7: Create a Database Connector.

Now you need to enter the connection details of the Database - namely, Database type, host, port, and Database name.

In our case, we will be using a MySQL Database that runs on a local machine (localhost). The port will be the default 3306, and we will use a Database named "qbtodb".

The system will then automatically generate the URL based on the information provided.

You can now test the connection by clicking on the *Validate* button. Once you have successfully validated the connection, you can save the connection values.

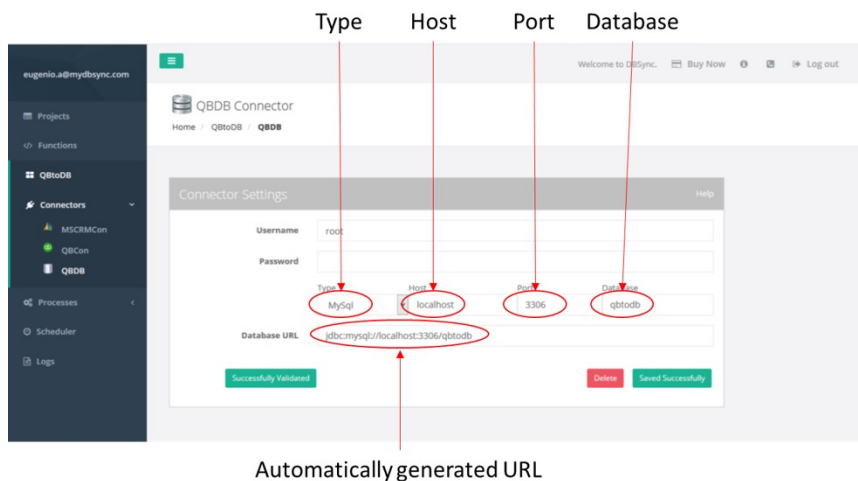


Figure 8: Database connection details.

Step 4: Create a Process

In order to create a new Process, click on the *Create New Process* button. The system will ask you for a name of your Process. Enter a name of your choice. Just, once more, remember that the name cannot include blank spaces. Then, click on the *Save* button. Your Process will be added to the list of Processes.

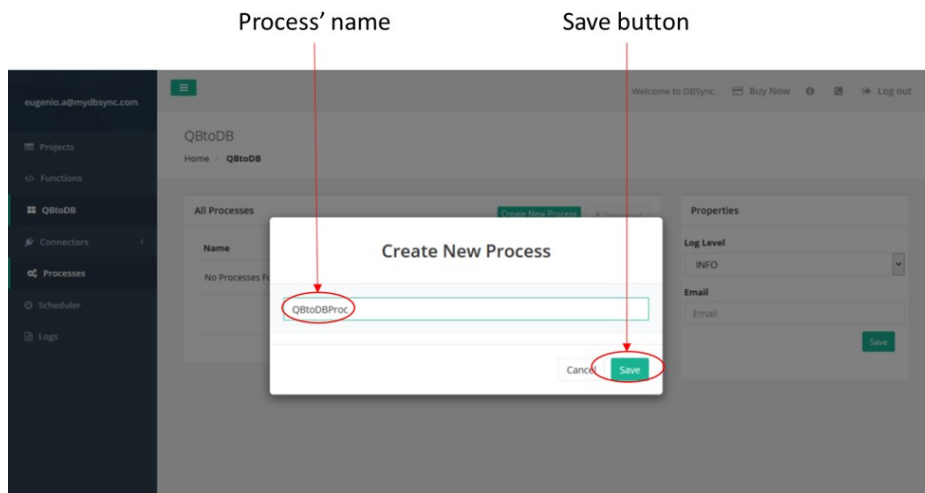


Figure 9: Create a new Process.

Step 5: Create a Workflow

Now that you have created a Process, you need to include at least one Workflow. Workflows define the data transfer i.e., what data is transmitted, the source and, the destination.

To create a Workflow, click on the *Create New Workflow* button. The system will ask you for a Workflow's name. Enter any name of your choice. Once again, remember that WorkflowProc names must not include blank spaces. Then, click on the *Save* button.

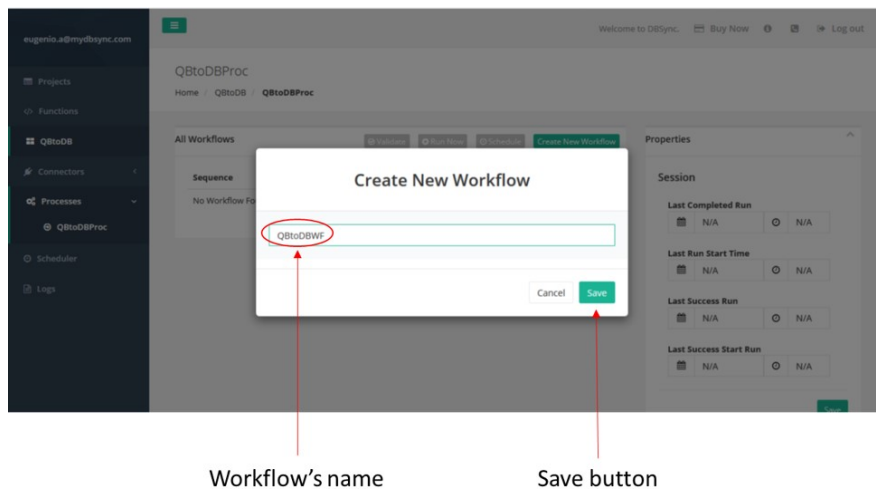


Figure 10: Create a new Workflow.

As explained earlier, Workflows include the data transfer information. Therefore, you need to define a data source or trigger; and, a data destination or rule. The rule must include a mapping between the source and the destination data.

Create a Trigger

In order to define a Trigger, you need to input the Data Source or Connector first. In this case, it would be the QuickBooks' Connector.

The system will now load all the objects available in that Data Source. Select one of them. In this example, we will use the "Account" object.

It is important to remember that you can have **only one** Data Source.

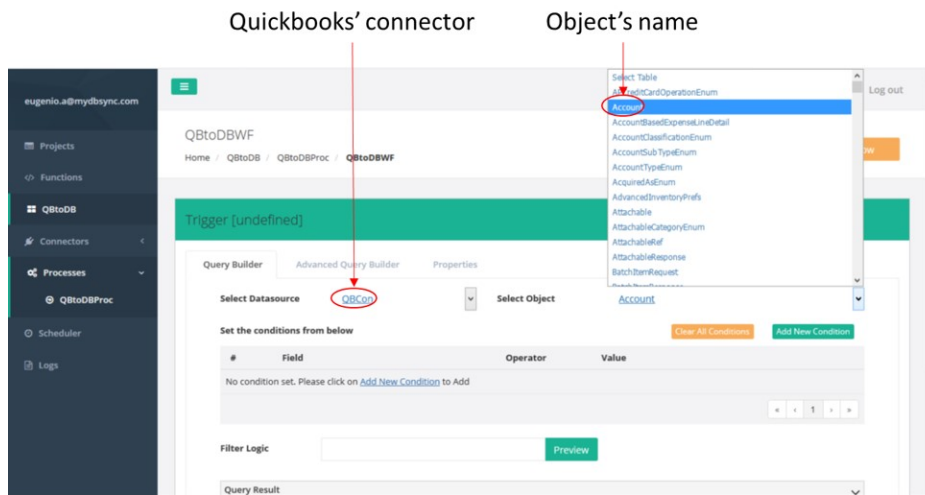


Figure 11: Create a Trigger.

Create a Rule:

Now, you need to define a data destination. This is done via a Rule definition. However, notice that now you can define more than one Rule or data destinations.

Click on the *Add New Rule* button. A new Rule line will appear in the list of Rules. You can either change the Rule name or, keep the default name.

Select the data destination Connector from the Menu. In this case, the MySQL Database Connector. Then, select an operation. Your choices are:

1. Insert: Inserts data from the source to the destination.
2. Update: Updates data on the destination, based on changes on the source.
3. Upsert: Updates and Inserts at the same time.
4. Delete: Deletes data on the destination - based on the information from the source.

Select a target object from the Menu. In our example, we have created a table named "Accounts" in the database "qbtodb".

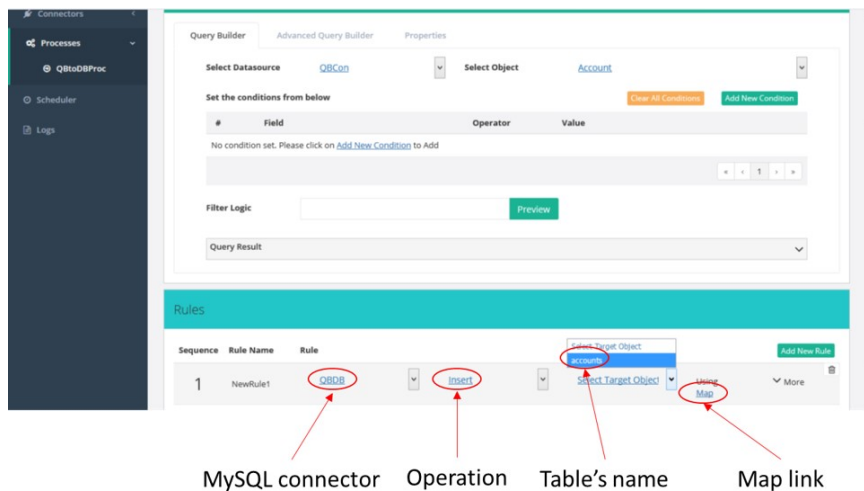


Figure 12: Create a Rule.

Create a Mapping

Up to this point, you have defined a data source and a data destination. The next step is to define the relationship between the two. In order to do so, click on the *Map* link. A screen showing both - data source objects and data destination objects - will appear.

Mapping fields on the *Mapping Screen* is very easy. You just need to drag and drop fields from the data source schema into the target or data destination list.

In this example, the QuickBooks fields "Name" and "Current Balance" are mapped to the Database fields - "AccountName" and "AccountAmount"

- respectively.

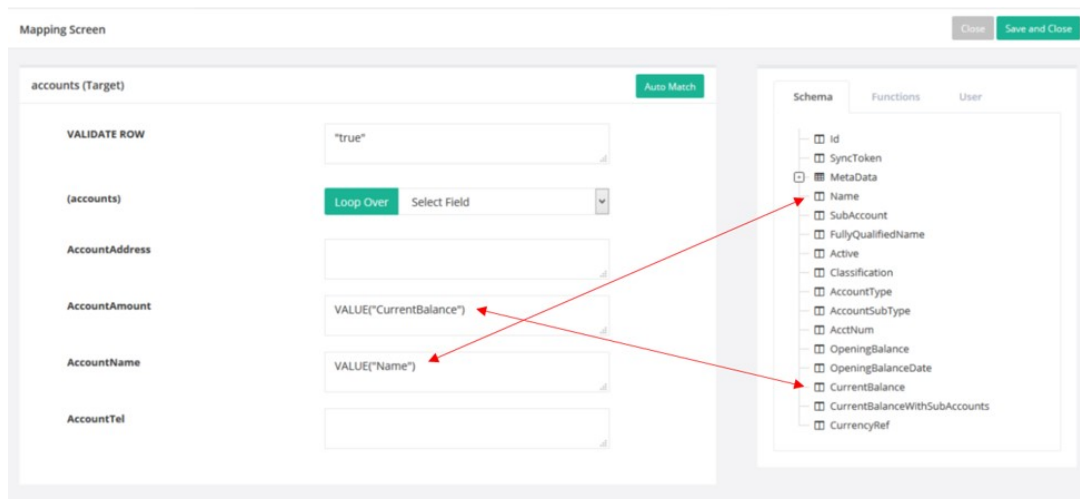


Figure 13: Mapping Screen

Save your Workflow

Although it may not look so important, this step is crucial. Because if you don't save the Workflow details, you will lose the information and have to redo all of it again.

To save the Workflow, simply click on the *Save Workflow* button. A message indicating success will appear. Click *OK*, and your Workflow will be added to the list of available Workflows in the process.

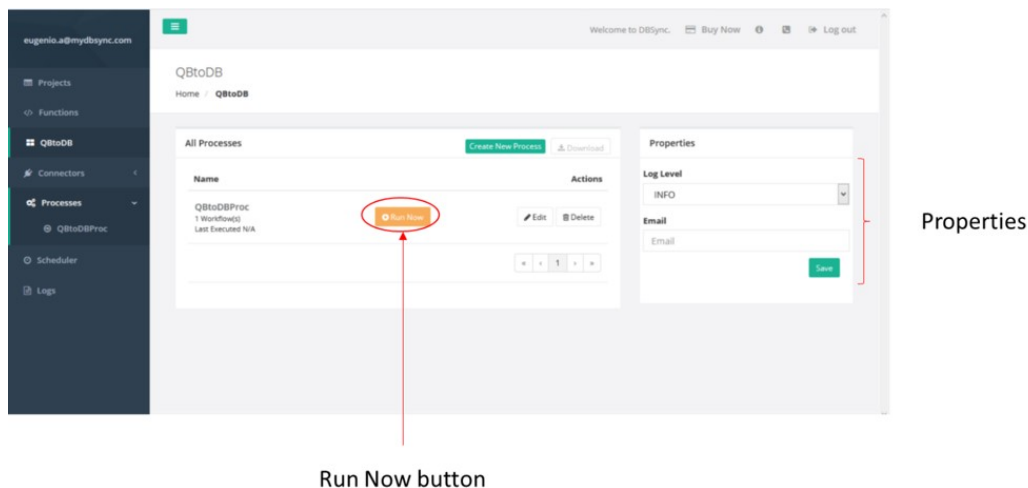
Figure 14: Save the Workflow.

Step 7: Run the Process

Figure 15 shows the screen with all the Processes available. The example has one Process with one Workflow.

This screen also shows a *Properties* section, where you can define the type of log level. And, add an Email so that the results of the run can be sent there.

In order to run the Process, click on the *Run Now* button. You can see the completed steps on the *Console* (figure 16).



Run Now button

Figure 15: Run the Process.

Console

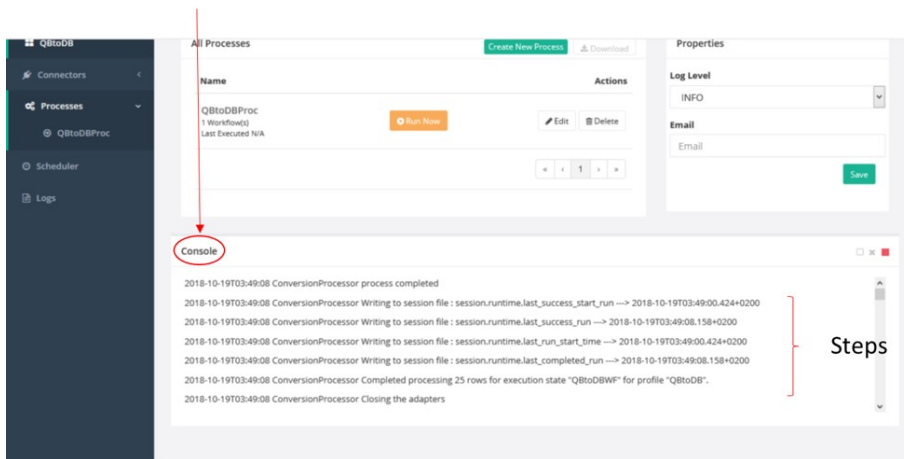
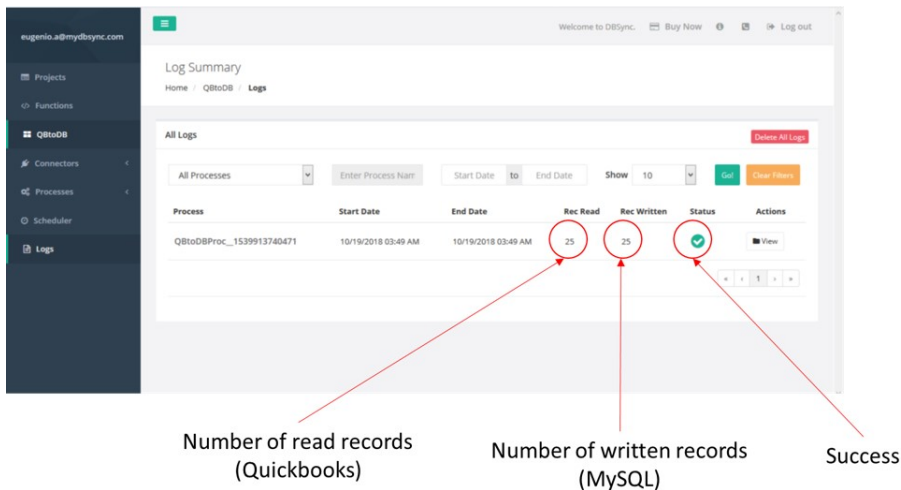


Figure 16: Console.

Step 8: Check the Logs

Once the run is done, you can go to the *Logs* section that appear on the menu on the left-hand side. The *Log Summary* shows information about the run - such as the number of records read from the source and the number of records written on the destination. There is also an icon showing success (green background, check symbol); or, failure (red background, exclamation symbol). There is also a link to the Logs. This can come handy in case you run into problems.

In our example, the Process was successfully run. And, 25 records were read from QuickBooks, and sent to the MySQL database.



Number of read records
(Quickbooks)

Number of written records
(MySQL)

Success

Figure 17: Log Summary.

Step 9: Check the Results

Now that you know that your Process was successfully run, it is wise to check the Database.

In our example, we open the "qbtodb" Database with the *phpMyAdmin* tool (figure 18), and verify that all the 25 records have been properly transferred.

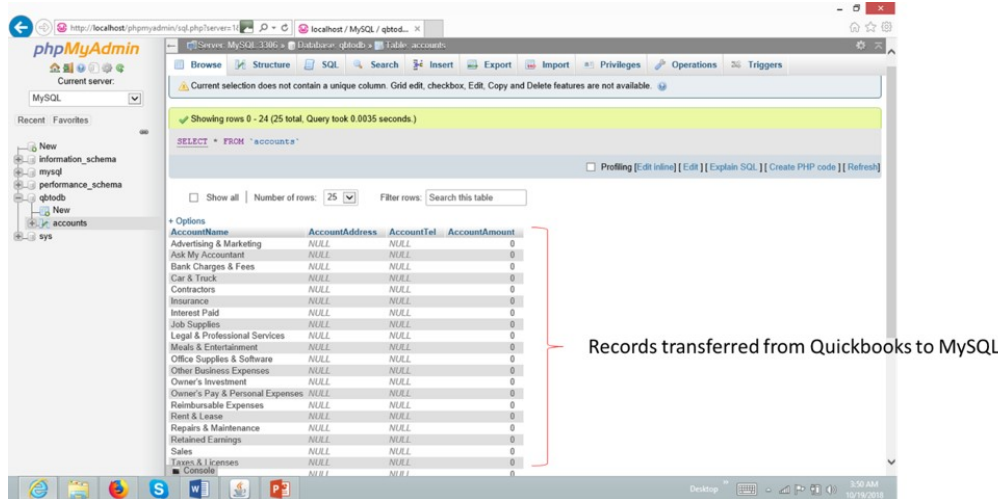


Figure 18: Database with records transferred from QuickBooks.

Scheduler

A useful feature of DBSync is its *Scheduler*. It allows you to automate your Processes by defining running schedules.

Creating a new schedule is straightforward. Select *Scheduler* from the left-hand side menu. The *Scheduler* screen will ask for several parameters - such as, Process Name, Start Date and Time and, a repeat routine that can be defined in minutes, hours, weekly, or at a specific day and time. The tool also accepts Cron expressions.

Once you have defined these values, you need to save the schedule by clicking on the *Save* button. Once successfully saved, the schedule will be added to the list of available schedules.

You can add as many schedules as you need.

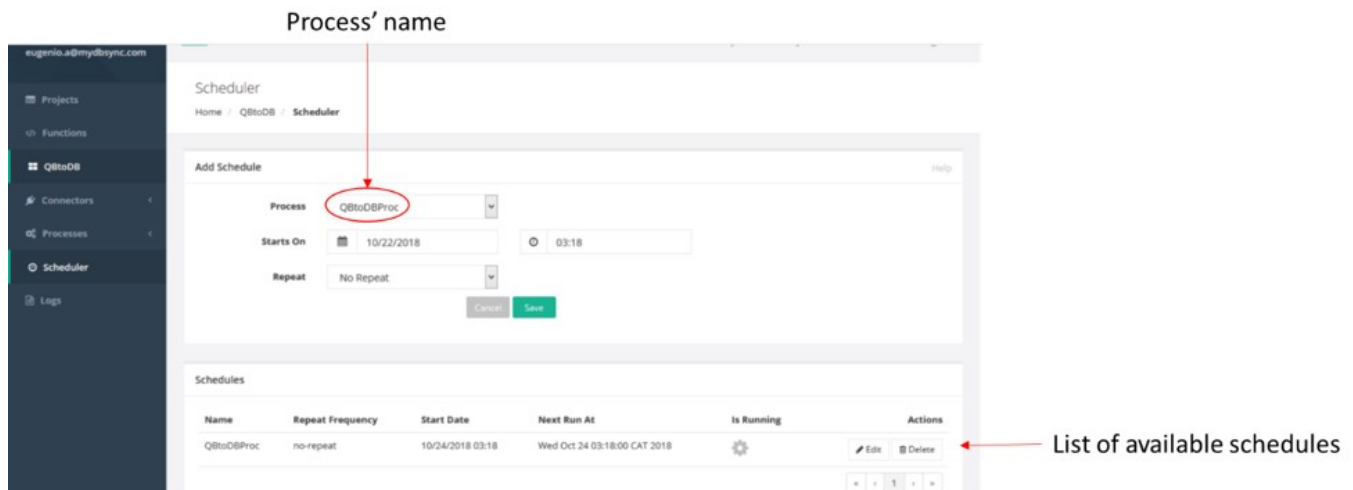


Figure 19: Scheduler

Final words

Throughout this series of three articles, you saw how to integrate QuickBooks with a Database, by using the DBSync application.

In the first part, you learnt how to create a Project and Connectors for QuickBooks and a Database respectively.

In the second part, you saw how to define your integration flow. That is, how to define a Process and a Workflow. A key aspect of this section was to define a mapping between QuickBooks objects and Database fields.

In this third and last section, you learnt how to run the Process and check that all values are properly saved in the Database.

The example showed how to transfer data from QuickBooks to a MySQL Database. Similarly, you could have transferred to another Database - such as Oracle or MSSQL. DBSync integration also works bi-directionally - meaning that you can transfer data from QuickBooks to a Database and vice versa.