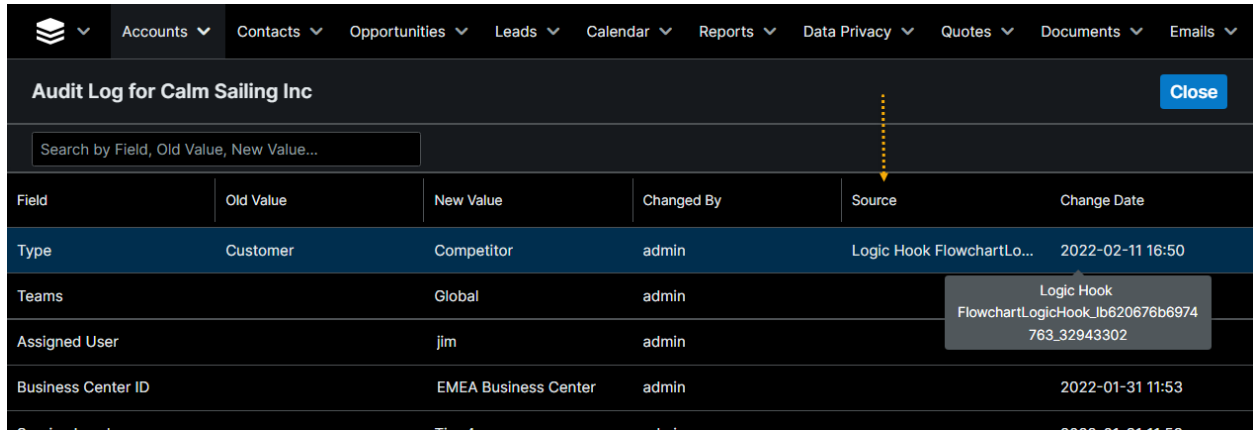


Flowchart unique name in Audit Log

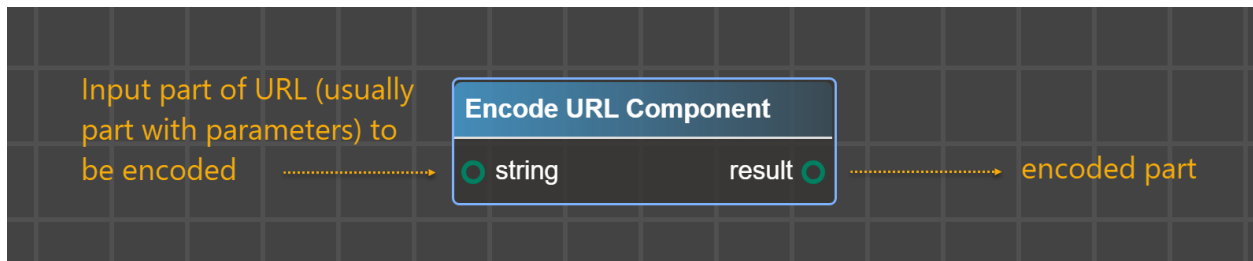
Whenever a flowchart changes a field, which is being audited, the flowchart name is displayed in the Audit Log:



Field	Old Value	New Value	Changed By	Source	Change Date
Type	Customer	Competitor	admin	Logic Hook FlowchartLo...	2022-02-11 16:50
Teams		Global	admin	Logic Hook FlowchartLogicHook_lb620676b6974763_32943302	
Assigned User		jim	admin		
Business Center ID		EMEA Business Center	admin		2022-01-31 11:53

New operators to ease connection via API

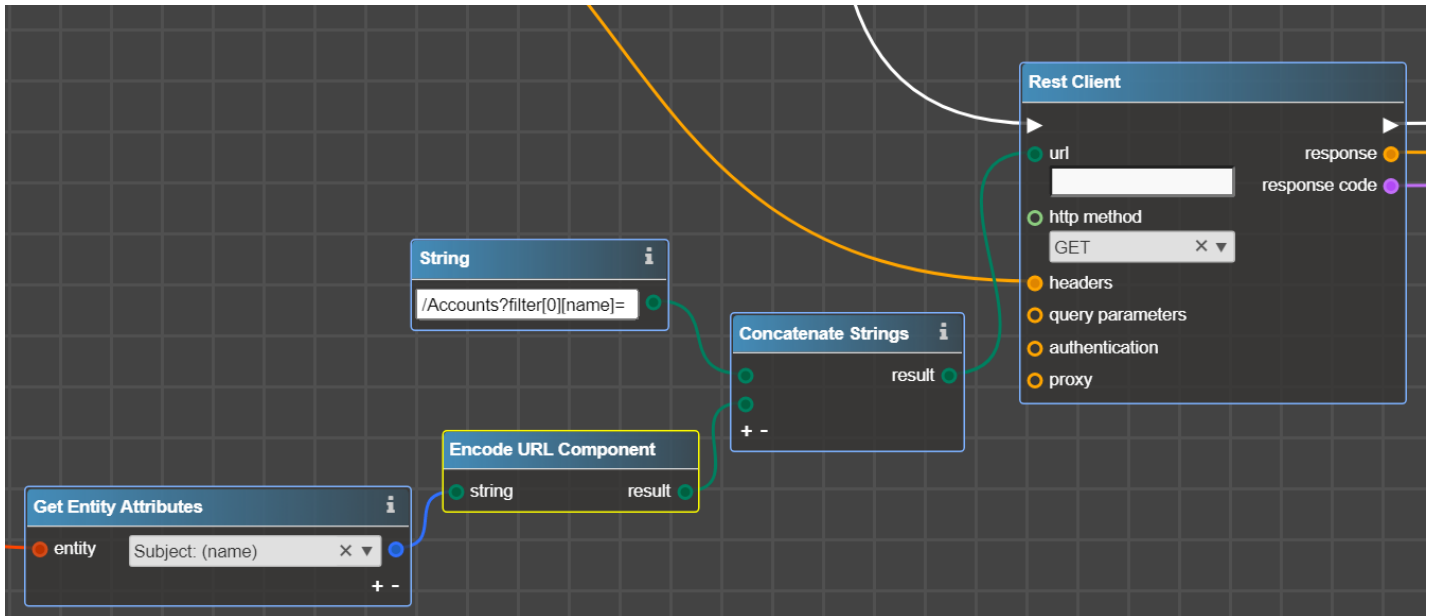
If you build an integration between Sugar and the 3d party system and provide parameters via URL, you can encode parameters with one operator **Encode URL Component**:



Encode URL Component converts reserved or non-ASCII characters passed in input into accepted ones by servers.

For example, “John Smith” is converted to “John %20Smith”

Example of usage where Account Name (which may contain spaces) is encoded prior to being put as a parameter into URL:



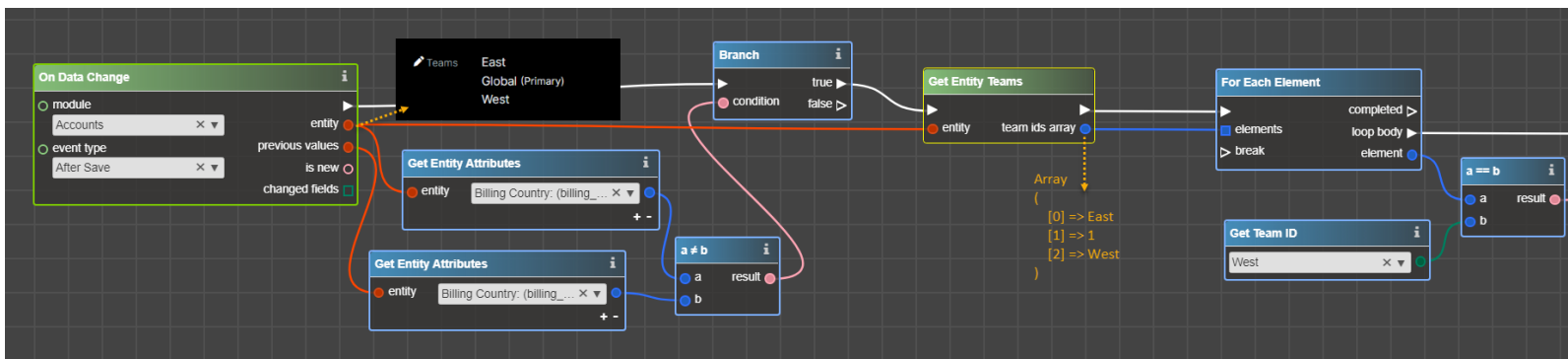
New operators for manipulating teams for Records

There are a bunch of new operators that allow you to read, add or remove teams for records.

Get Entity Teams

Now you can get all Teams that are assigned to the record with an operator **Get Entity Teams**. Useful to check whether there is a particular Team set for the record.

Just provide *entity* and as output there is an **Array** of all Teams assigned to the record:



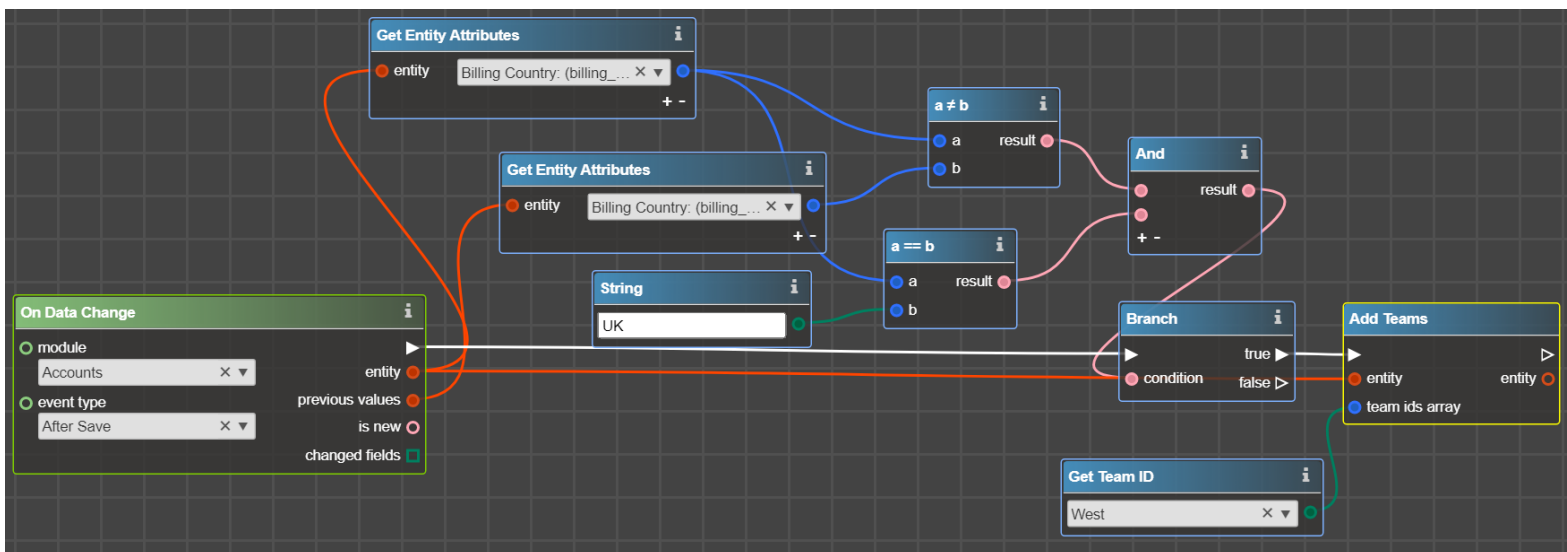
Several notes that apply to the following operators of Teams manipulating:

1. In Sugar each teamset for record consists of two components: `team_id` as Primary Team and `team_set_id` as all other teams. New operators work with `team_set_id` only. See chapter [Manipulating Primary Team](#) to find out how to work with `team_id` aka Primary Team.
2. The recommended approach is to use new operators on After Save flowcharts than on Before Save.
3. Do not use **Save Entity** to preserve team changes. Each of the new operators applies team changes itself.
4. Despite any changes that might be applied with new operators, the standard Sugar logic, which ensures access of Assigned User to a record, works. Let's say that you created flowcharts and because of its execution a teamset for the record do not contain any Team, to which an Assigned user belongs. In this case, standard Sugar behavior adds Assigned Users' Private Team.

Add Teams

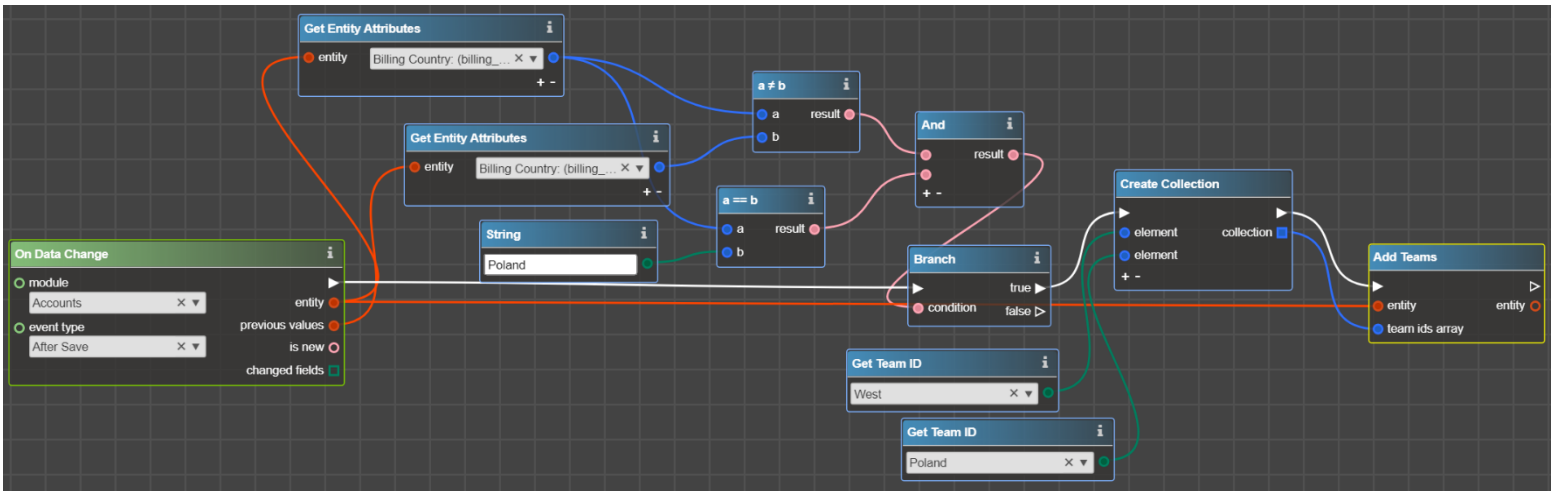
It does what it says – adds Team or several Teams to the record.

To add one Team simply provide an ID of a Team to add an *entity* for which Team(s) must be added:



As a result of the flowchart execution team West is added to the set of teams of Account Record.

To add several Teams, provide an **Array** of Team IDs or **Collection** of Team IDs add and entity for which Team(s) must be added:

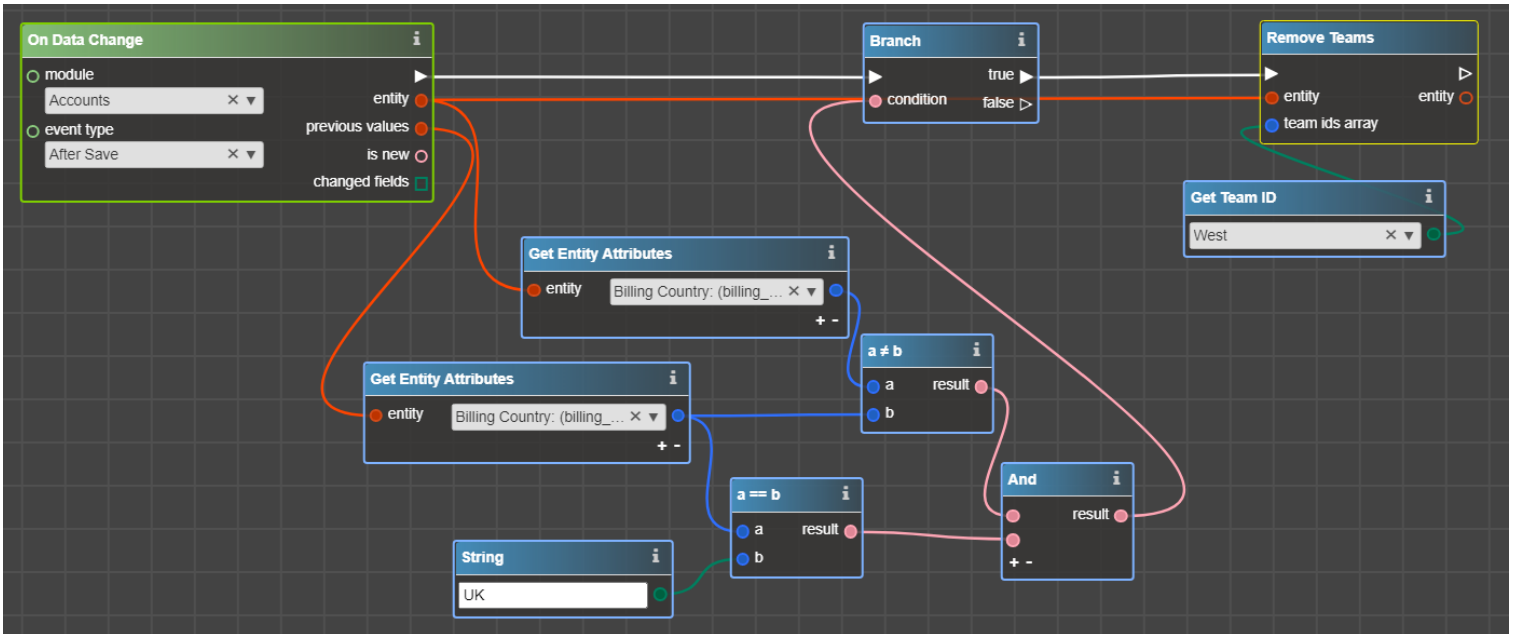


As a result of the flowchart execution two teams West and Poland are added to the set of teams of Account Record.

Remove Teams

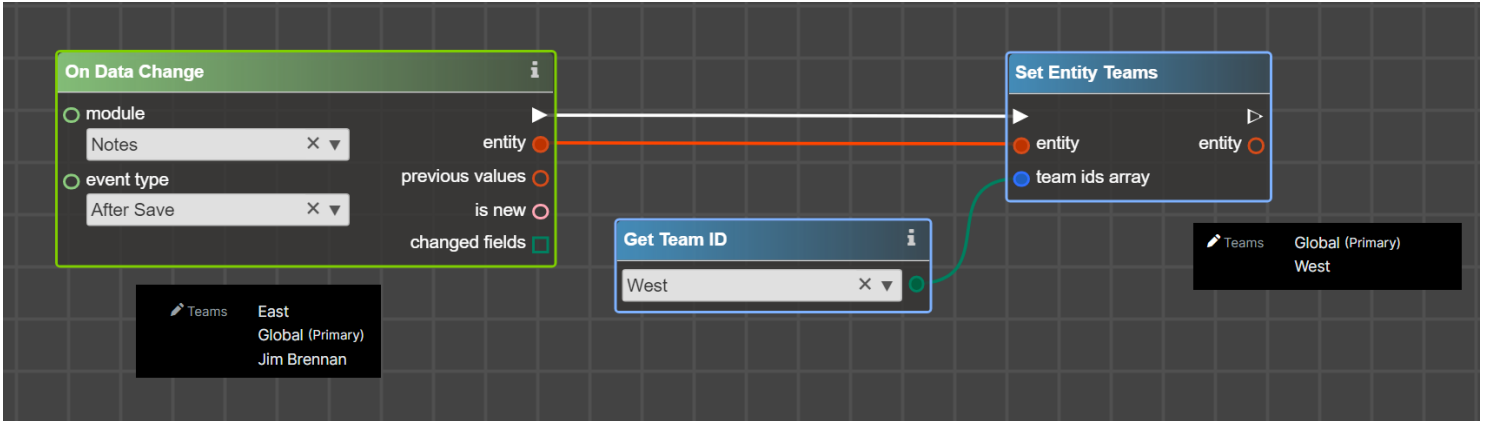
Conversely to the operator **Add Teams**, this operator removes Team or Teams from the record.

It works in the same way as **Add Teams** – it expects **entity** from which Teams should be removed and **Collection, Array, or one Team ID** to be removed:



Set Entity Teams

Completely overrides `team_set_id` with the teams provided in the input. In the same way as the previous two operators, this one expects an *entity* where Teams must be changed and *Array*, *Collection*, or one team ID that must be set as `team_set_id`.



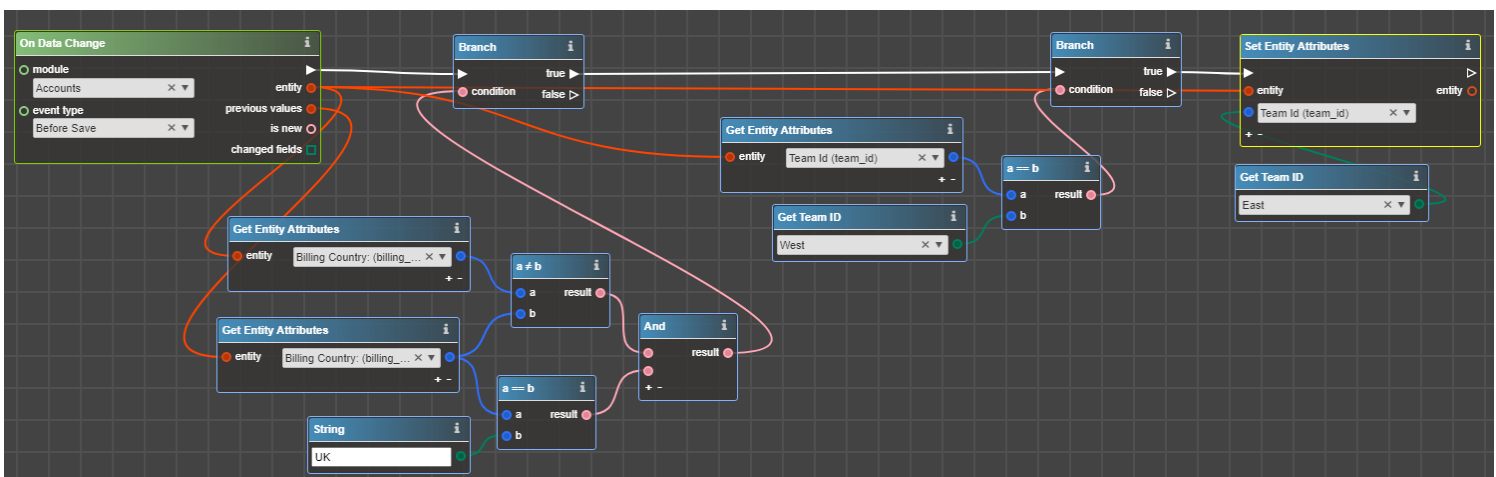
Manipulating Primary Team

As mentioned above, new operators of Teams manipulation work only with `team_set_id`. However, sometimes you will have to consider Primary Team (or `team_id`) in your flowcharts as well.

You can work with a Primary Team as with the usual field in Logic Builder using **Get/Set Entity Attributes** on Before Save.

For example, when it is needed to remove a Team based on conditions and you do not know where this Team was set – as Primary or not, this challenge could be addressed with two flowcharts:

- first one on Before Save ensuring that Team will be removed from the Primary. Please note that the Primary Team must be always set:

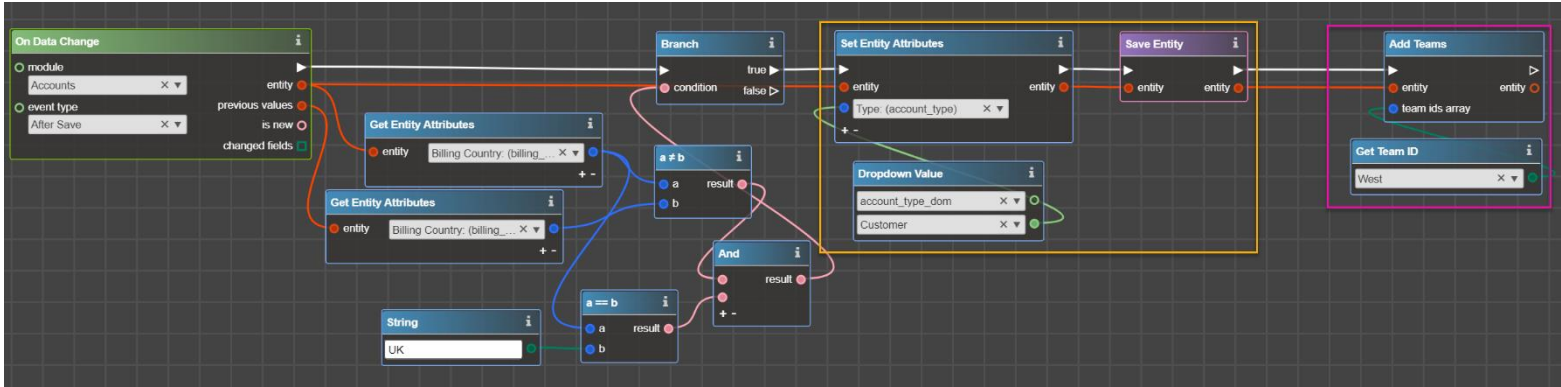


- and the second flowchart on After Save to remove Team for the case when this Team is not Primary (see section [Remove Teams](#)).

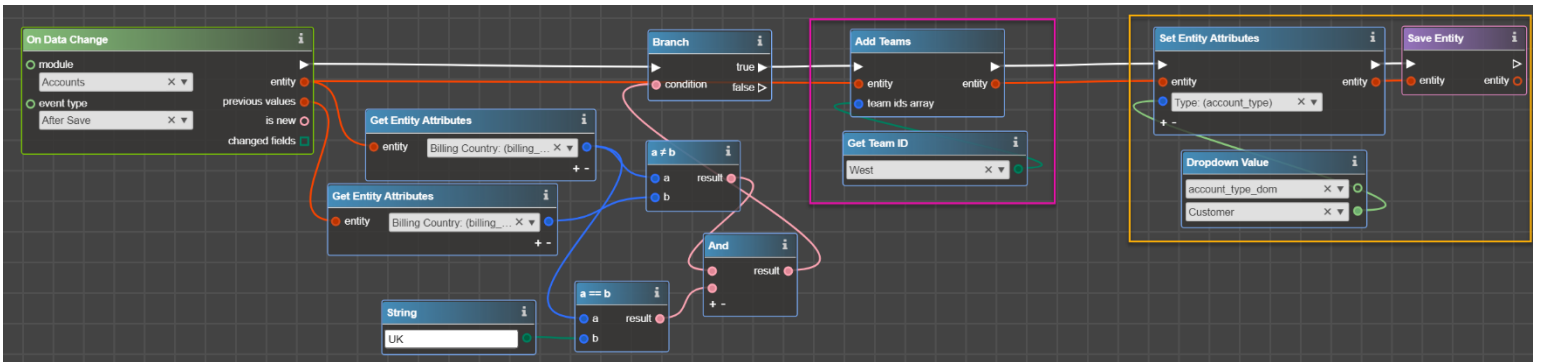
I need to update an entity and change teams; how should I do this?

In one After Save flowchart.

If you add (or remove) Teams and then update the record, any other flowchart that works on team change (for example, if you have implemented logic of teams inheriting from Account to related records) will be triggered.



Otherwise – firstly, update the record, then update the Team – other flowcharts that work on team change won't be triggered.



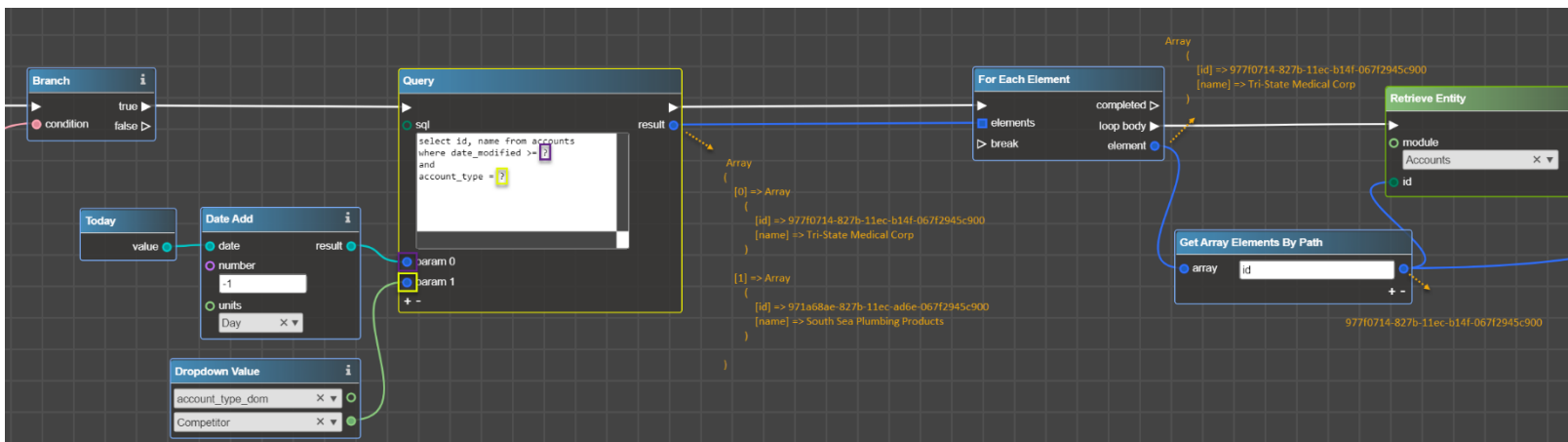
New operator to perform Direct SQL Requests to the Database

Now you can write your own SQL queries with operator **Query**.

To use it input SQL request directly in the operator. Use the symbol "?" for parameters in your query and input parameters values in the operator. Symbols "?" are filled with provided values in order they are inputted. As a result there is:

- Array of arrays of returned records for *select* queries
- 1 (true) or 0/empty (false) for updates/inserts

- Error text in case of issues with query



Fixed Issues:

1. Fixed unexpected results of comparison operators when there is a comparison with 0.
2. Added spaces trim in **Get/Set Array Element**, **Get/Set Array Elements by Path**.
3. Fixed error when the latest deployed Service was called instead of the enabled one.
Please update **LB Services Extension** and **LB SugarBPM Extension** to resolve this issue.
Also, flowcharts On Service Call should be re-generated and re-deployed on the instance if you encounter this error.