



Onepoint Ltd Talend Kudu Components







Contents

| | |
|--------------------------------------|----|
| Introduction | 3 |
| Abstract | 3 |
| About Apache Kudu | 3 |
| Pre-Requisites | 3 |
| Kudu Installation | 3 |
| Talend Installation | 3 |
| Talend Components Folder Setup | 3 |
| Kudu Components Installed | 4 |
| Support Materials | 5 |
| Example Schema | 5 |
| tKuduOutput | 5 |
| Example Job 1 | 5 |
| Step by step instructions | 5 |
| Example Job 2 | 10 |
| Step by step instructions | 10 |
| tKuduInput | 18 |
| Example Job 1 | 18 |
| Step by step instructions | 18 |
| Example Job 2 | 22 |
| Step by step instructions | 22 |
| Common Errors..... | 27 |
| Requested Replication Factor | 27 |
| Solution | 27 |
| Connection Failure | 27 |

INTRODUCTION

ABSTRACT

In this tutorial you can learn how to use the Talend Kudu components created by One point Ltd. These components are:

| Name | Description |
|---|---|
|  tKuduInput | This is the component used to read data from Apache Kudu. |
|  tKuduOutput | This is the component used to save data from Apache Kudu. |

These components are free and can be downloaded from [Talend Exchange](#).

ABOUT APACHE KUDU

Apache Kudu is a revolutionary distributed columnar store for **Hadoop** that enables the powerful combination of fast analytics on fast data. Kudu complements the existing Hadoop storage options, **HDFS** and Apache **HBase**. Additional information on Apache Kudu, its architecture and use cases can be found at (<http://getkudu.io/>).

At the time of this creation of this document (June 2016) the Apache Kudu is still in beta stage. Onepoint Ltd is planning to release a new version of the components as soon as Apache Kudu 1.0 is released.

PRE-REQUISITES

Kudu Installation

You will need to have Apache Kudu installed in order to be able to use the components. Apache Kudu runs on multiple Linux distributions and can be installed following the instructions on this page:

<http://getkudu.io/docs/installation.html>

A developer friendly option to be able to develop on one single machine would be to use a Cloudera VM with Linux on which you run Kudu and then have Talend running on the hosting OS.

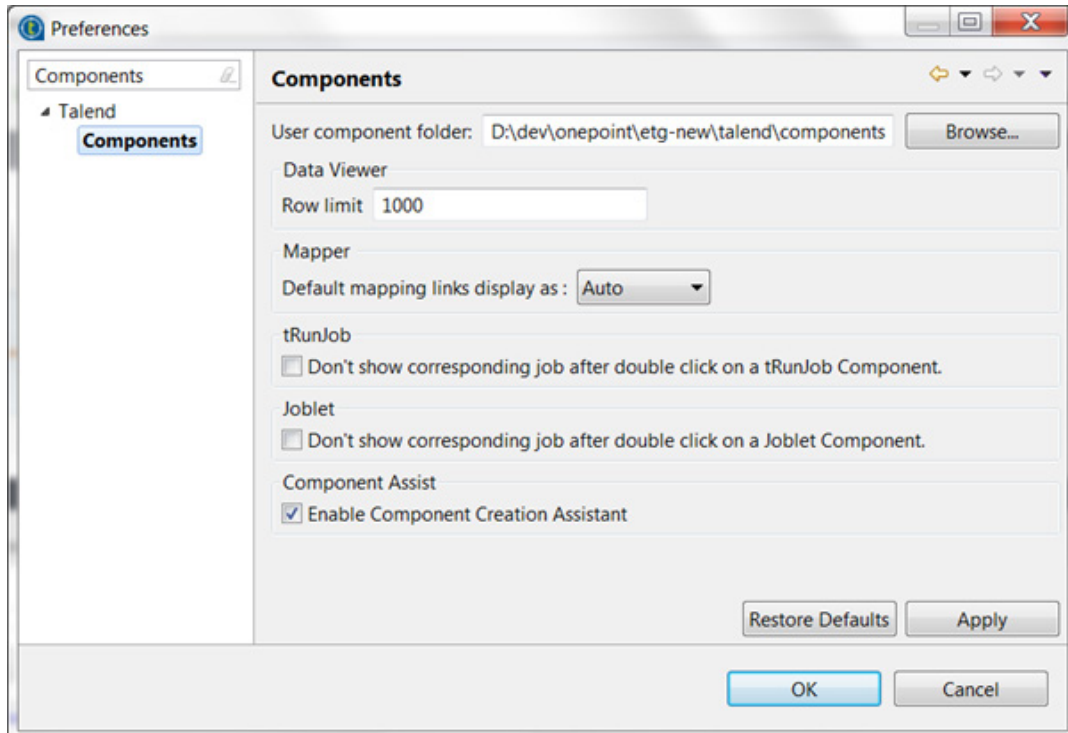
Talend Installation

You will also need to have at least Talend Open Source 6.0 installed on your machine, in order to be able to use the components. Any of the Talend Enterprise versions would of course also work for this tutorial.

Talend Components Folder Setup

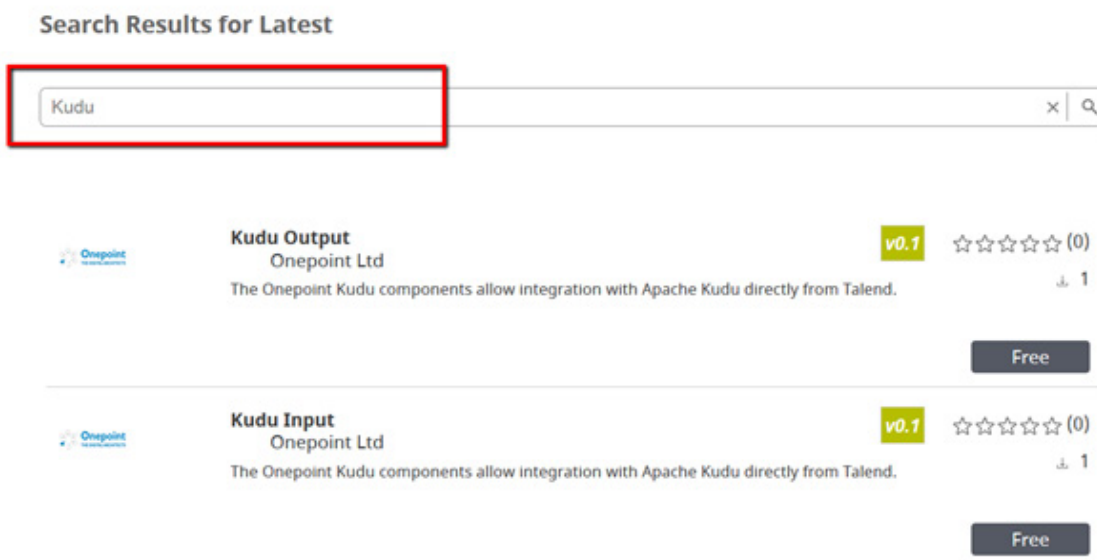
Finally you will need to have the components folder properly setup, so that you can install the components from [Talend Exchange](#). Here are the instructions to do so:

<https://help.talend.com/display/KB/Installing+a+custom+component>



Kudu Components Installed

Finally you should have the Kudu components installed in your Talend Components folder. The easiest way to find the components in Talend Exchange is simply by searching for "Kudu":



Support Materials

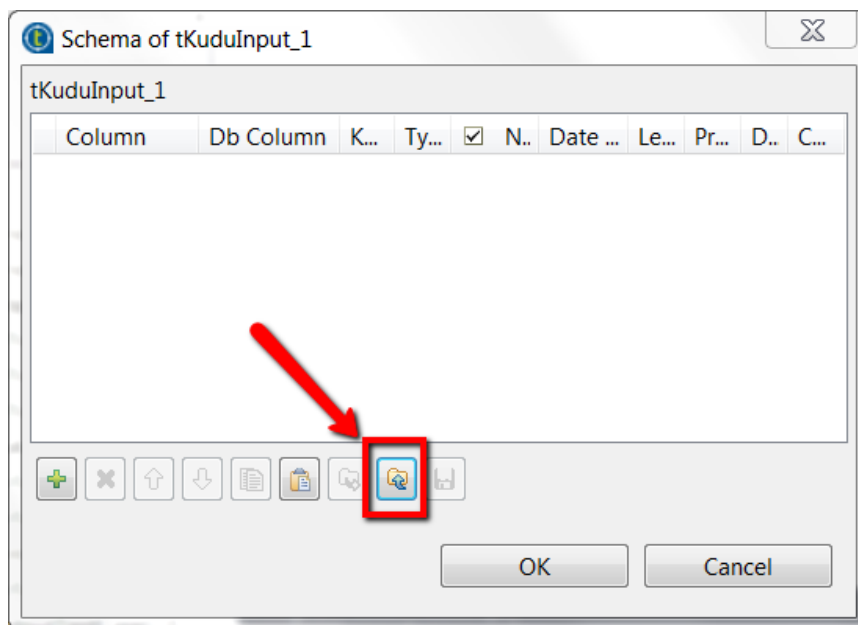
EXAMPLE SCHEMA

The schema used in the examples is always the same. It represents the data of a customer and might be tedious to create manually. For this reason we provide an xml export of the schema which you can use in this tutorial.



kudu_tutorial_schema.xml

In order to import the schema into any of the components mentioned in the examples, please use this button:



tKuduOutput

This component allows you to write data to Apache Kudu. It accepts one input flow connection. Furthermore it also supports optional output and reject flow connections.

Optionally the component allows you to create and delete Kudu tables too.

EXAMPLE JOB 1

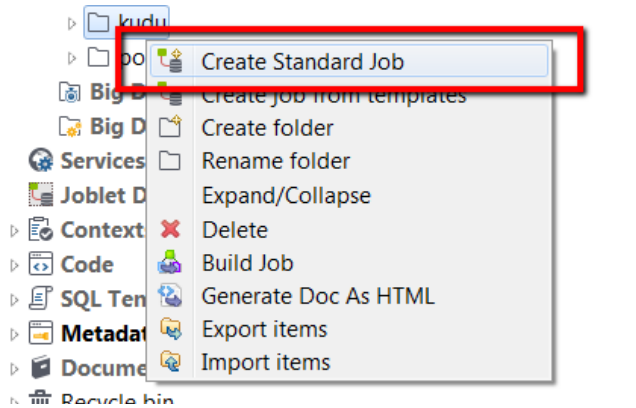
In this job we will write some dummy data to a Kudu table which will be created in case the Kudu table does not exist yet.

Step by step instructions

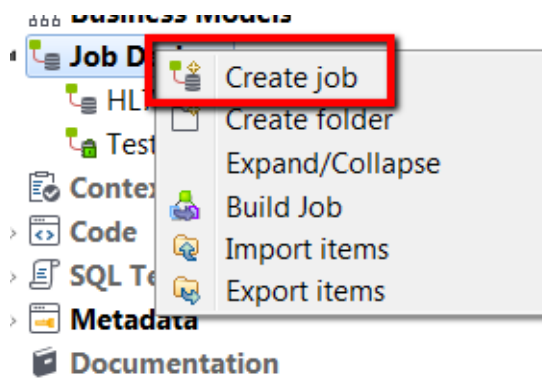
1. We will start by creating a standard Talend job (if you are using the "Enterprise version"). If you are using the open source version of Talend you just typically create a normal job.



a. Enterprise version



b. TOS version



2. We will fill the details of the New Job dialogue.

New Job
Add a job in the repository

Name: KudoTutorial1
Purpose: Show how tKuduOutput works
Description: Show how tKuduOutput works
Author: user@talend.com
Locker:
Version: 0.1 [M] [m]
Status:
Path: kudu [Select]

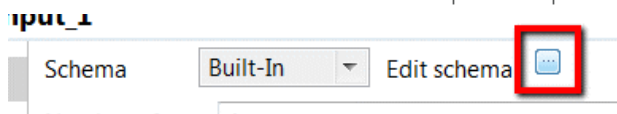
[Finish] [Cancel]



3. We select the tFixedFlowInput component from the Palette and drop it on the job view panel.

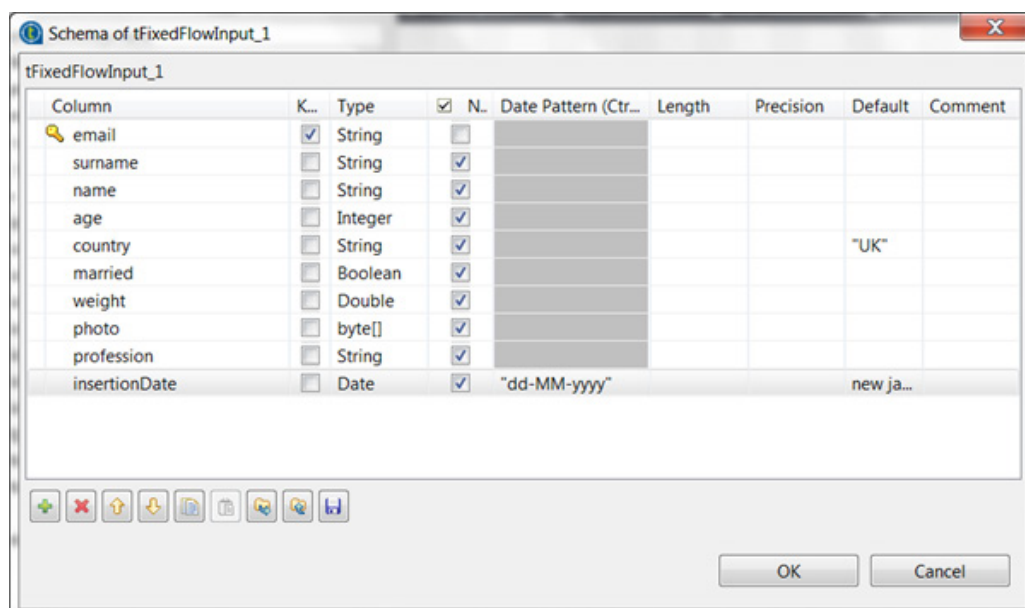


4. We click on the created tFixedFlowInput component and click on the "Edit schema" button.



5. The schema we are going to create describes a customer. It contains the following fields:

- a. Email (the primary key)
- b. Surname
- c. Given name
- d. Age
- e. Country
- f. Married
- g. Weight
- h. Photo
- i. Profession
- j. Insertion Date

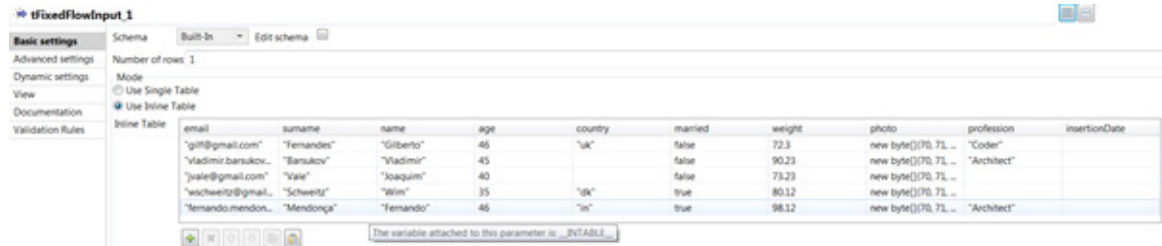




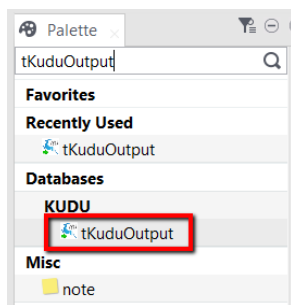
Please note that Kudu always needs a primary key which is in this case the email field.

Hint: alternatively you can import the schema file provided in this tutorial (see chapter Support Materials).

6. Now we create the data for this same component. For this purpose we are going to use an inline table.



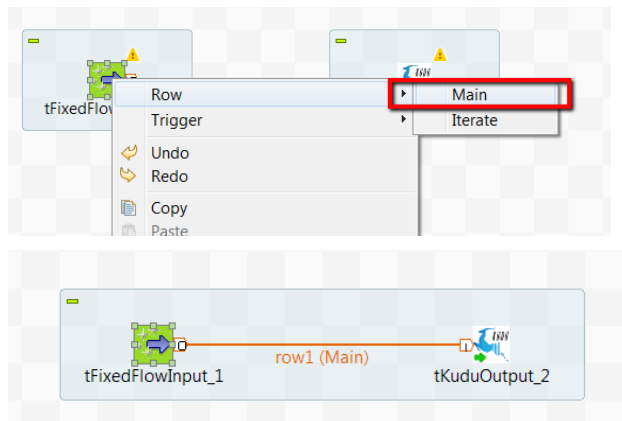
7. At this point in time we have a fully configured tFixedFlowInput component which can be linked to a tKuduOutput component. Now we search in the palette for the tKuduOutput component which you can typically find in the category "Databases/Kudu".



8. We select the tKuduOutput component from the Palette and drop it on the job view panel.



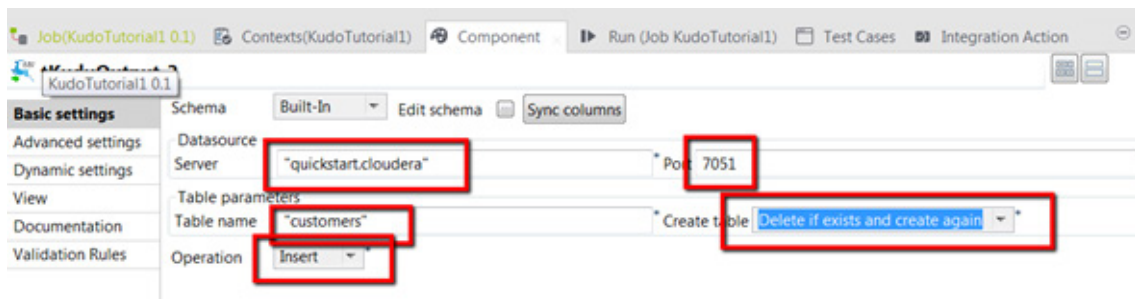
9. Now we connect the tFixedFlowInput component with the tKuduOutput component.



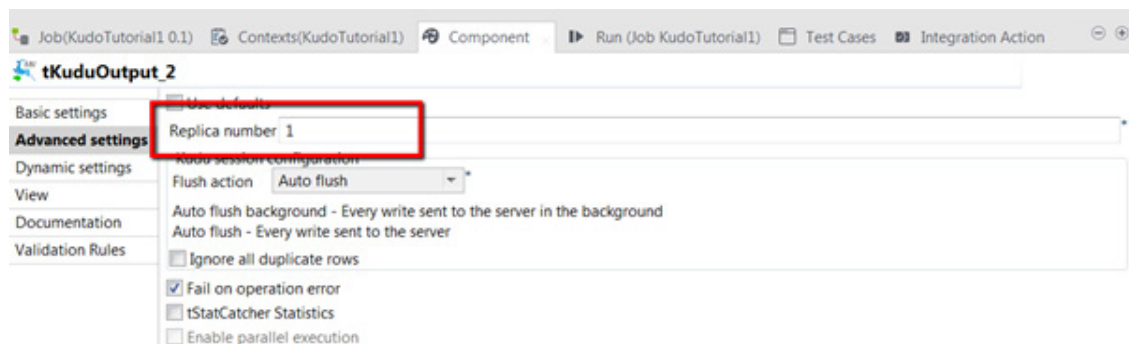


10. The tKuduOutput connection needs to be configured. We click on the tKuduOutput component and change the data in the “Basic settings” view. You have to set all parameters on this panel:

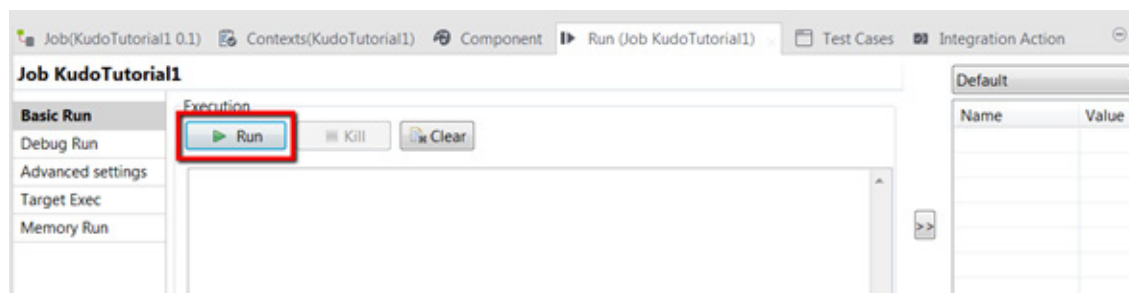
- a. Server – The name of the server on which Apache Kudu is running. Please note that on test environments you might have to change the hosts file to map the name to a specific IP address.
- b. Port – The port on which Apache Kudu is running.
- c. Table name – The name of the table which is going to store the data.
- d. Create table – The table creation options. We have chosen “Delete if exists and create again”, because we want to guarantee that this example runs without errors.
- e. Operation – The data operation to be executed by this component. In this case we are going to insert data.



11. (Optional) If you have started Kudu on a Cloudera distribution VM or on a simple VM, most probably you will need to set the number of replicas to 1.

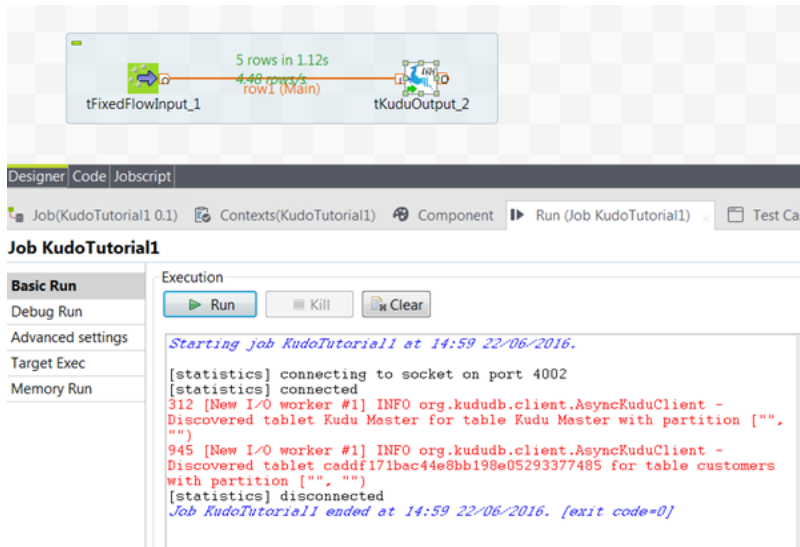


12. Now we can run the job and see, if everything is ok.





13. In case of success you should see something like this on Talend Studio:



In case of errors, please check the Common Errors chapter

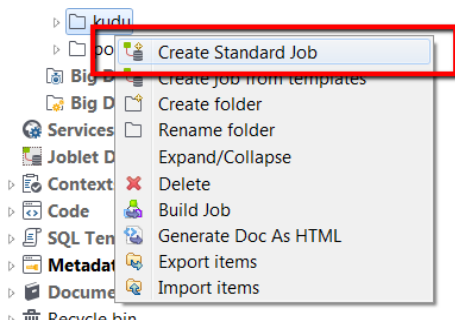
EXAMPLE JOB 2

In this job we will write some dummy data to a Kudu table. Some of this data will be correct and some of this data will violate the primary key contract and will be rejected.

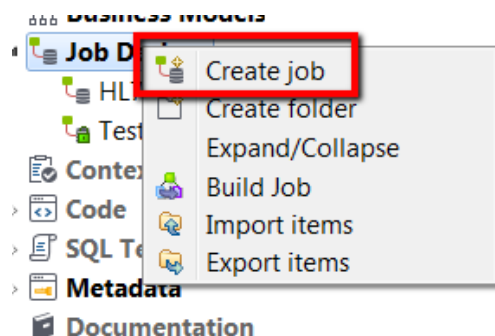
Step by step instructions

1. We will start by creating a standard Talend job (if you are using the "Enterprise version"). If you are using the open source version of Talend you just typically create a normal job.

a. Enterprise version

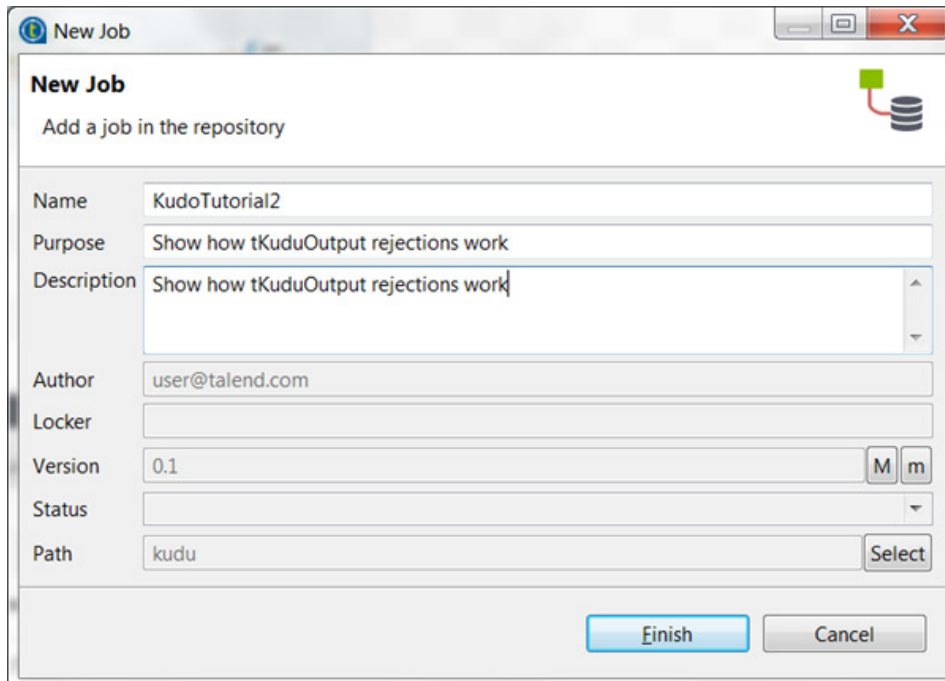


b. TOS version

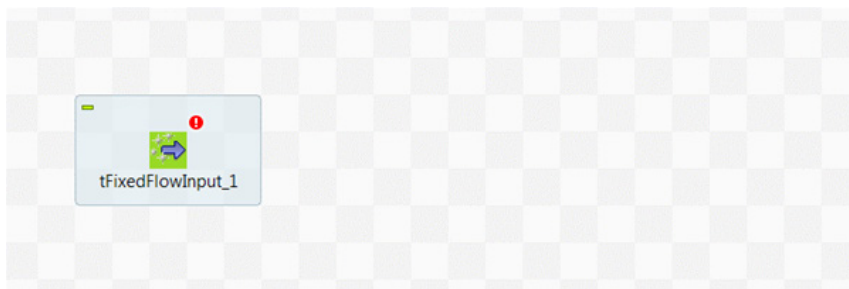




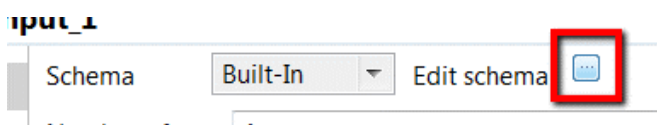
2. We will fill the details of the New Job dialogue.



3. We select the tFixedFlowInput component from the Palette and drop it on the job view panel.



4. We click on the created tFixedFlowInput component and click on the "Edit schema" button.

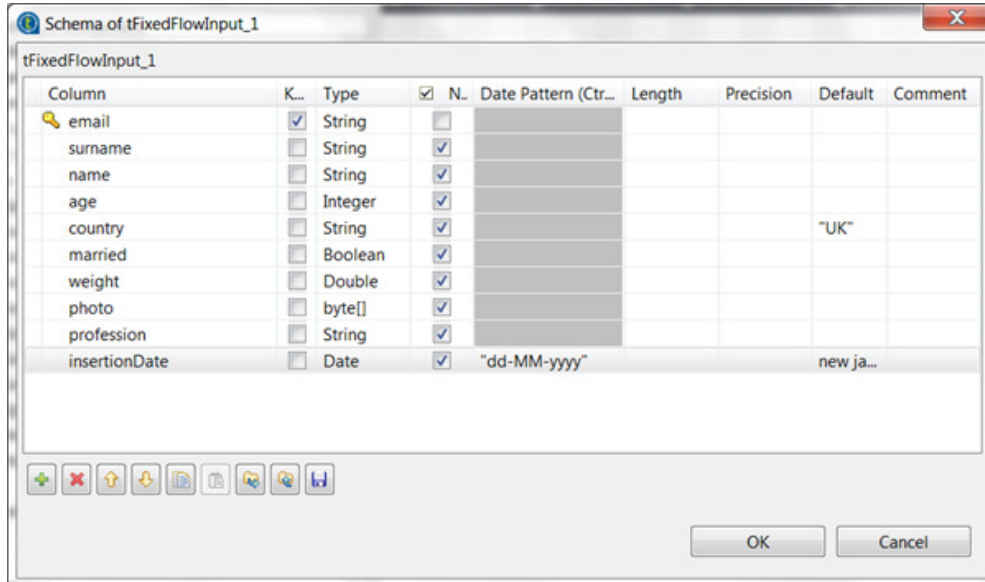


5. The schema we are going to create describes a customer. It contains the following fields:



- a. Email (the primary key)
- b. Surname
- c. Given name
- d. Age
- e. Country
- f. Married
- g. Weight
- h. Photo



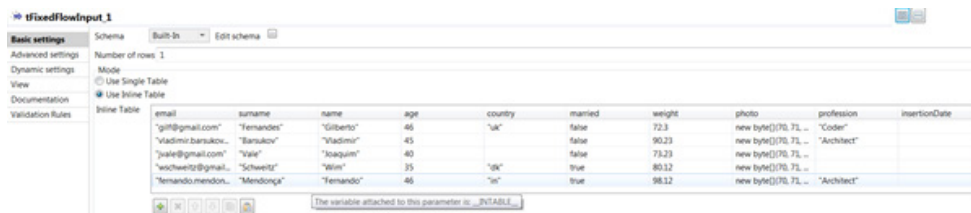
- i. Profession
- j. Insertion Date





Please note that Kudu always needs a primary key which is in this case the email field.

If you have completed the first job in this tutorial you can simply copy / paste the schema fields using the copy / paste buttons ( ). Or you can simply import the schema file provided in this tutorial (see chapter Support Materials).

6. Now we create the data for this same component. For this purpose we are going to use an inline table.



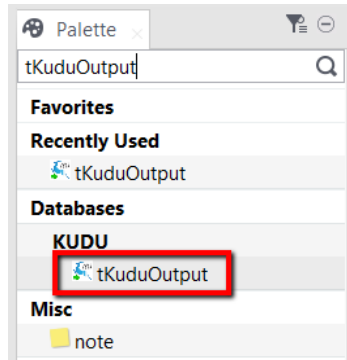
If you have completed the first job in this tutorial you can simply copy / paste the data fields using the copy / paste buttons ( ).

7. Now we are going to duplicate the first row of the tFixedFlowInput inline table component. We





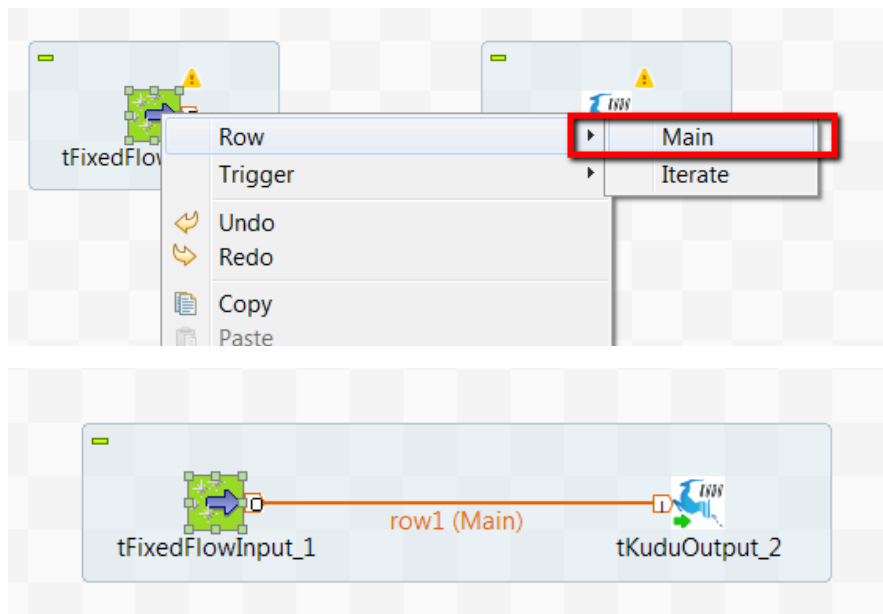
8. At this point in time we have a fully configured tFixedFlowInput component which can be linked to a tKuduOutput component. Now we search in the palette for the tKuduOutput component which you can typically find in the category “Databases/Kudu”.



9. We select the tKuduOutput component from the Palette and drop it on the job view panel.



10. Now we connect the tFixedFlowInput component with the tKuduOutput component.

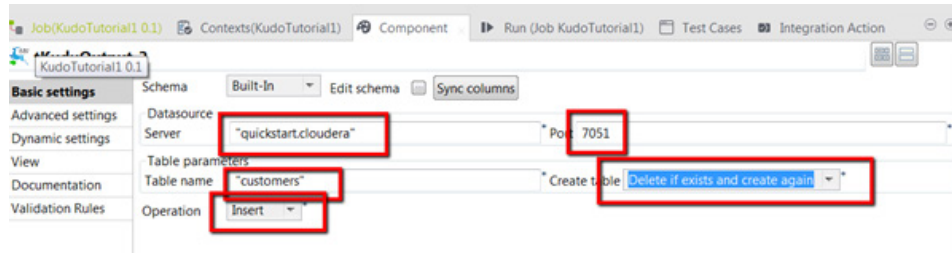


11. The tKuduOutput connection needs to be configured. We double-click the tKuduOutput component and change the data in the “Basic settings” view. You have to set all parameters on this panel:

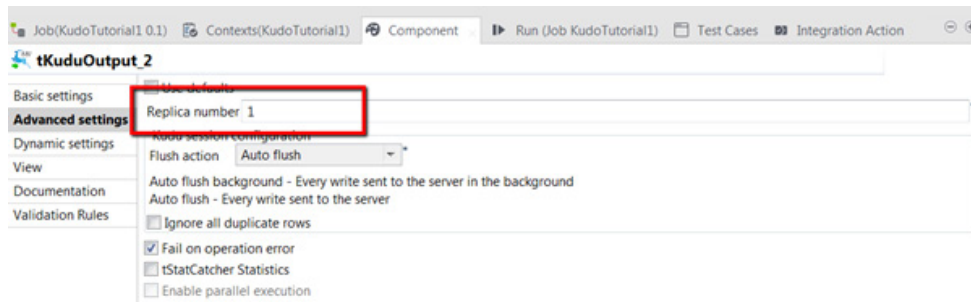
- a. Server – The name of the server on which Apache Kudu is running. Please note that on test environments you might have to change the hosts file to map the name to a specific IP address.



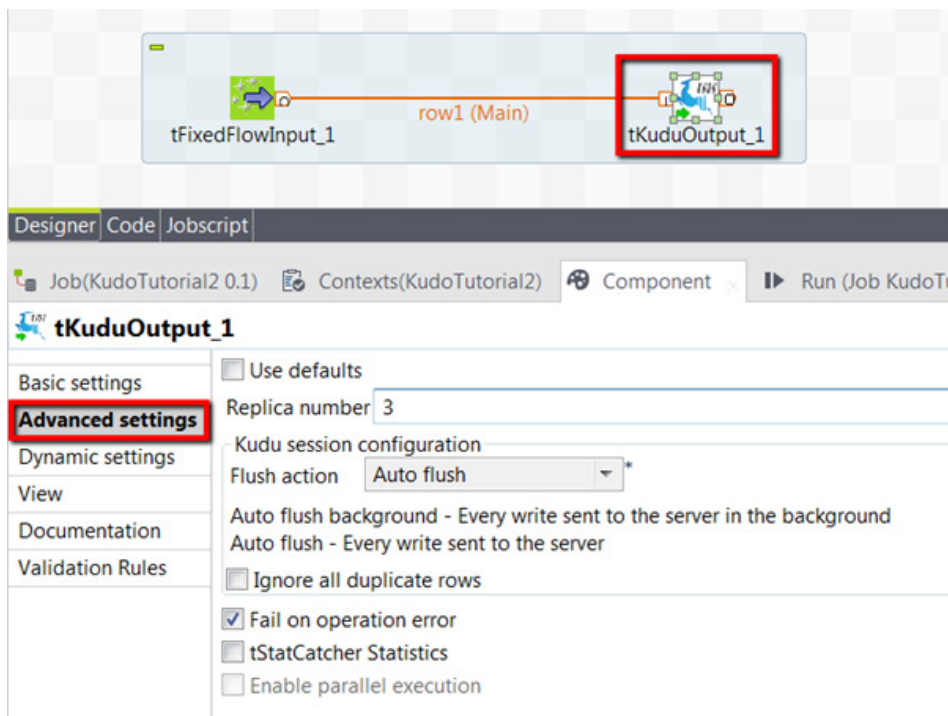
- b. Port – The port on which Apache Kudu is running.
- c. Table name – The name of the table which is going to store the data.
- d. Create table – The table creation options. We have chosen “Delete if exists and create again”, because we want to guarantee that this example runs without errors.
- e. Operation – The data operation to be executed by this component. In this case we are going to insert data.



12. (Optional) If you have started Kudu on a Cloudera distribution VM or on a simple VM, most probably you will need to set the number of replicas to 1.

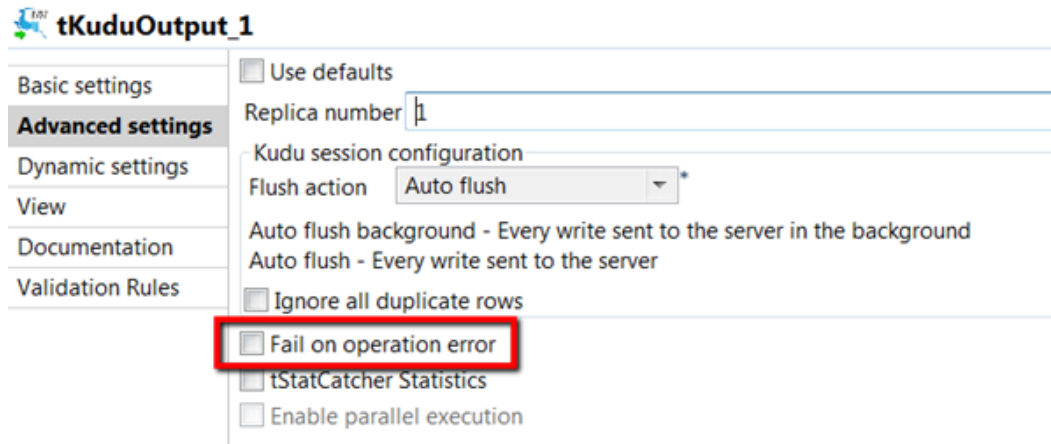


13. Now double-click on the tKuduOutput component and select the “Advanced Settings” tab.





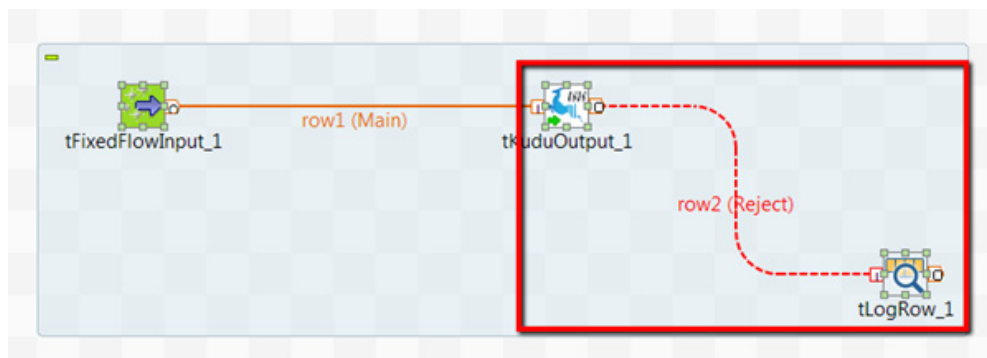
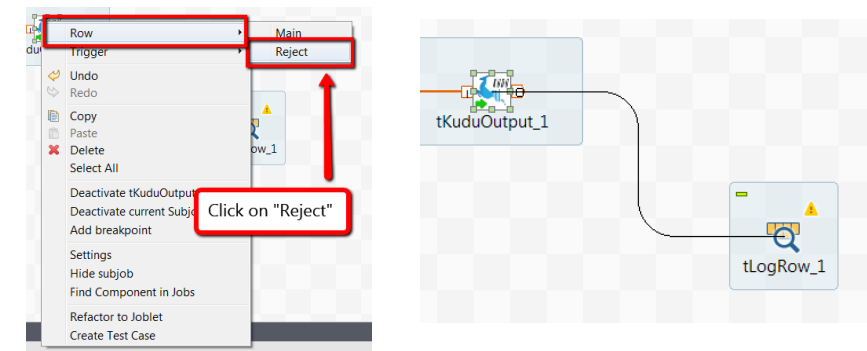
14. Now untick the "Fail on operation error" checkbox.



15. Now search in the palette for a tLogRow, select it and drop it on the job view panel.

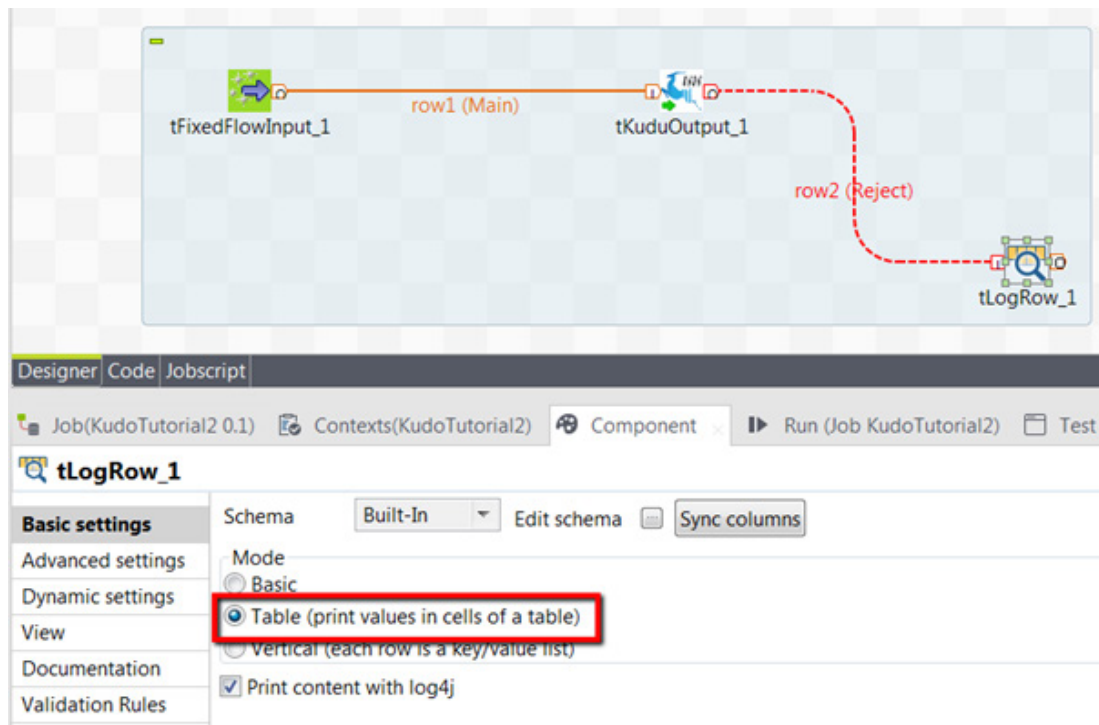


16. Now click with the right mouse on the tKuduOutput component and select "Reject". After that, drag the reject connector onto the tLogRow component.

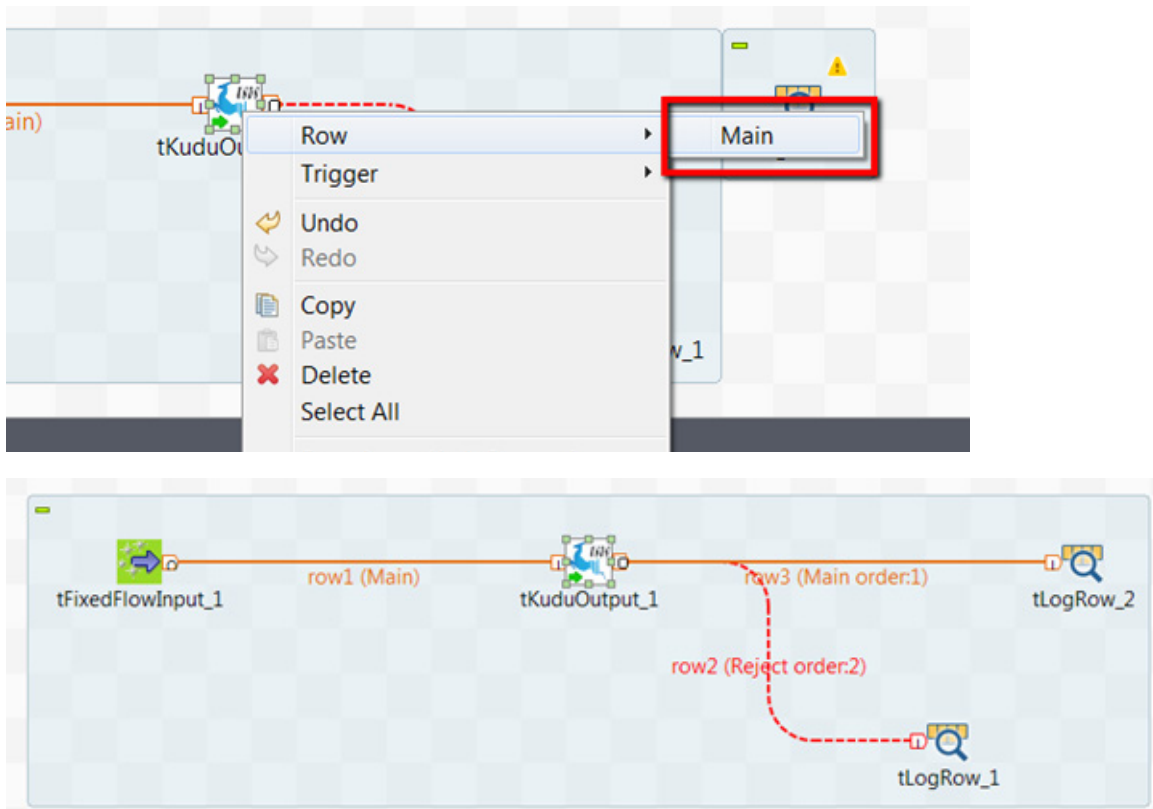




17. Now you can double click on the tLogRow component and select the "Table" mode.



18. Now we are going to add another tLogRow to this job and connect the tKuduOutput component to it using a regular row connector.





19. Also double-click on the tLogRow_2 component and please choose the "Table" mode.

The screenshot shows a Talend job configuration with the following components and data flow:

- tFixedFlowInput_1**: 6 rows in 1.18s (5.09 rows/s), row1 (Main)
- tKuduOutput_1**: 5 rows in 1.18s (4.24 rows/s), row3 (Main order:1)
- tLogRow_2**: 5 rows in 1.18s (4.24 rows/s), row3 (Main order:1)
- tLogRow_1**: 1 rows in 1.18s (0.85 rows/s), row2 (Reject order:2)

The **tLogRow_2** component settings are shown below:

- Schema: Built-In
- Mode: **Table (print values in cells of a table)** (selected)
- Vertical (each row is a key/value list): Unselected
- Print content with log4j: Selected

20. Now we run the job and if everything goes well, you should see that most of the rows except one are printed out by the tLogRow_2 component. One row will be rejected though, due to a duplicate primary key.

The screenshot shows the job execution results for **KudoTutorial2**. The execution log displays the following error:

```
990 [main] ERROR components_development.kudotutorial2_0_1.KudoTutorial2 - Row error for primary key="gilf@
```

The data output from the **tLogRow_2** component is shown in a table format:

| email | surname | name | age | country | married | weight | photo | profession | insertDate |
|-----------------------------------|-----------|----------|-----|---------|---------|--------|-------|------------|------------|
| gilf@gmail.com | Fernandes | Gilberto | 46 | uk | false | 72.3 | FGH | Coder | 23-06-2016 |
| vladimir.polev@gmail.com | Polev | Vladimir | 45 | UK | false | 90.23 | FGH | Architect | 23-06-2016 |
| jvale@gmail.com | Vale | Joaquim | 40 | UK | false | 73.23 | FGH | null | 23-06-2016 |
| wschwartz@gmail.com | Schwartz | Wim | 35 | dk | true | 80.12 | FGH | null | 23-06-2016 |
| fernando.mendonca@onepointltd.com | Mendonça | Fernando | 46 | in | true | 98.12 | FGH | Architect | 23-06-2016 |

The error message indicates that the row with email 'gilf@gmail.com' was rejected because it already exists in the table.

tKuduInput

This component allows you to read tabular data from Apache Kudu tables. You can either scan through the whole table or you can use query filters. This tutorial contains two example jobs, one demonstrating a scan and another one demonstrating how the end user can use the query fields.

Warning: you should have executed before proceeding either Example Job 1 or Example Job 2.

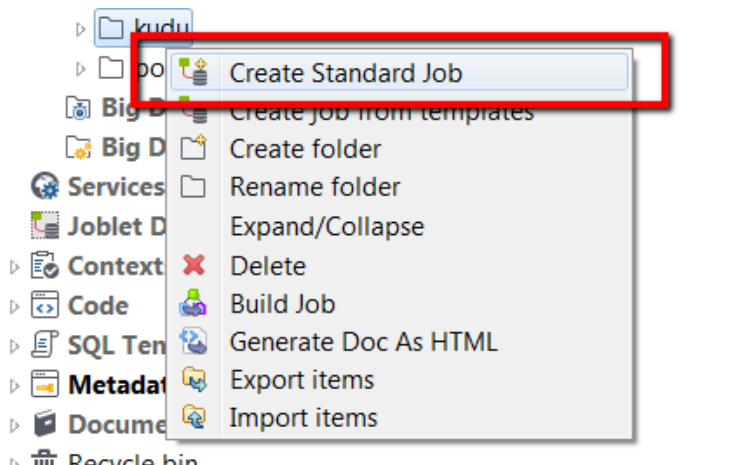
EXAMPLE JOB 1

In this example job you will learn how to setup the tKuduInput component and how to perform a full table scan on a Kudu component.

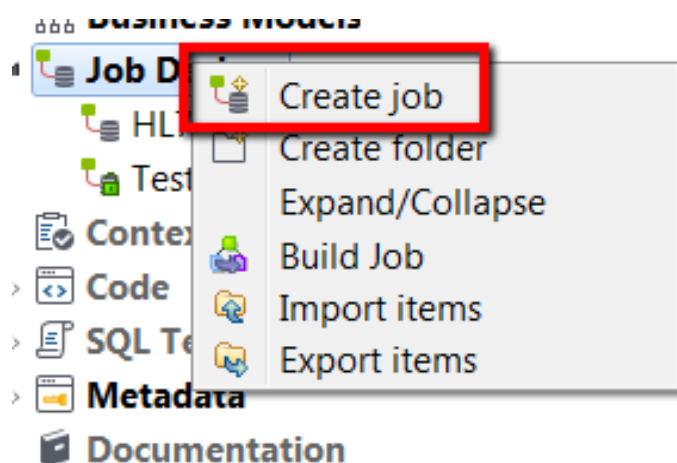
Step by step instructions

1. We will start by creating a standard Talend job (if you are using the "Enterprise version"). If you are using the open source version of Talend you just typically create a normal job.

a. Enterprise version

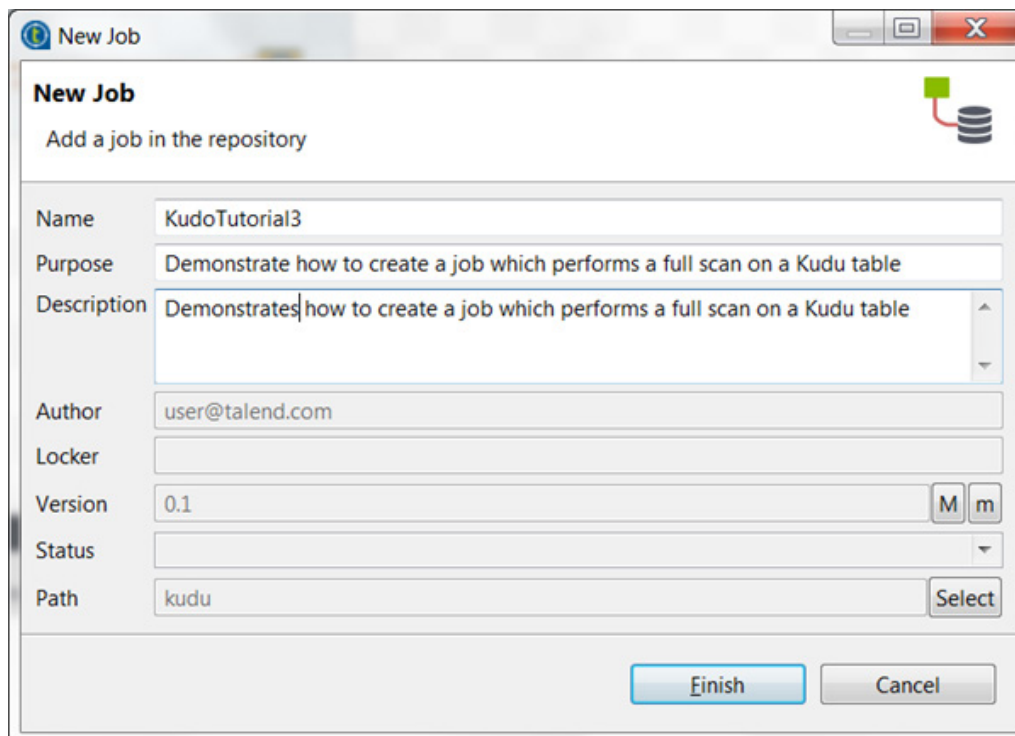


b. TOS version

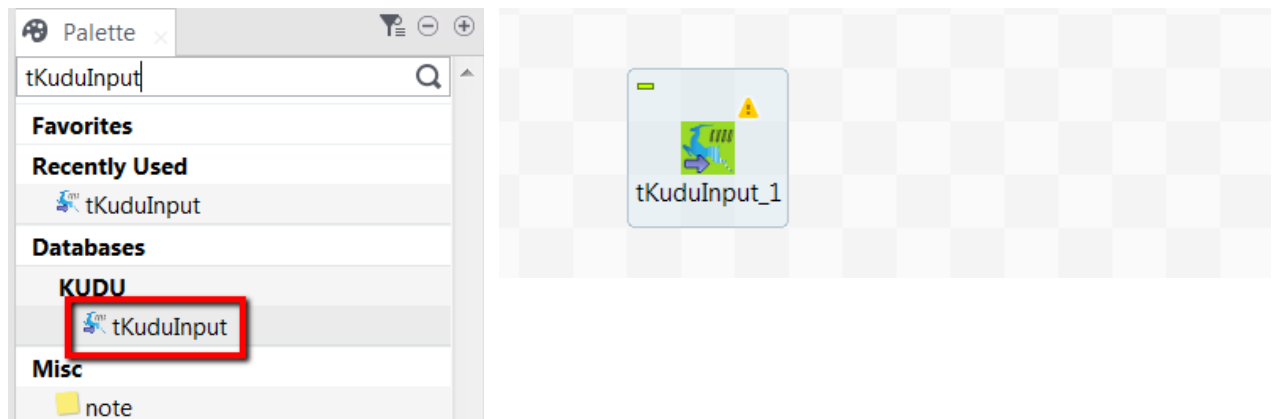




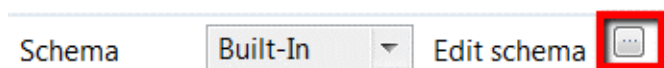
2. Here are the details of the created job.



3. We select the tKuduInput component from the Palette and drop it on the job view panel.



4. We double-click on the created tKuduInput component and click on the "Edit schema" button.



5. The schema we are going to create describes a customer. It contains the following fields:

- a. Email (the primary key)
- b. Surname
- c. Given name
- d. Age
- e. Country
- f. Married



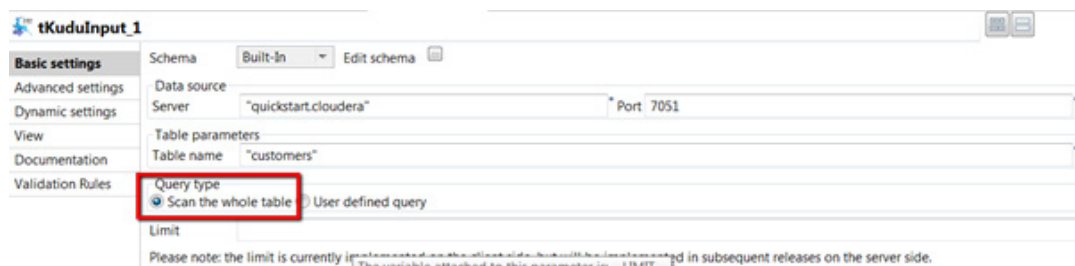
- g. Weight
- h. Photo
- i. Profession
- j. Insertion Date

| Column | Db Column | K. | Type | N. | Date Pattern (Ctrl+Sp... | Length | Precision | Default | Comment |
|---------------|---------------|-------------------------------------|---------|-------------------------------------|--------------------------|--------|-----------|---------|-------------|
| email | email | <input checked="" type="checkbox"/> | String | <input type="checkbox"/> | | | | | |
| surname | surname | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | | |
| name | name | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | | |
| age | age | <input type="checkbox"/> | Integer | <input checked="" type="checkbox"/> | | | | | |
| country | country | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | | "UK" |
| married | married | <input type="checkbox"/> | Boolean | <input checked="" type="checkbox"/> | | | | | |
| weight | weight | <input type="checkbox"/> | Double | <input checked="" type="checkbox"/> | | | | | |
| photo | photo | <input type="checkbox"/> | byte[] | <input checked="" type="checkbox"/> | | | | | |
| profession | profession | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | | |
| insertionDate | insertionDate | <input type="checkbox"/> | Date | <input checked="" type="checkbox"/> | "dd-MM-yyyy" | | | | new java... |

Hint: if you have already gone through the previous example jobs (Example Job 1, Example Job 2) you can simply copy the schema from the tFixedInputFlow component. Or you can simply the schema file provided in this tutorial (see chapter Support Materials).

6. The tKuduInput component needs to be configured. We double-click the tKuduInput component and change the data in the "Basic settings" view. You have to set all parameters on this panel:

- a. Server – The name of the server on which Apache Kudu is running. Please note that on test environments you might have to change the hosts file to map the name to a specific IP address.
- b. Port – The port on which Apache Kudu is running.
- c. Table name – The name of the table which is going to store the data.
- d. Query type – The selected value should be "Scan the whole table".

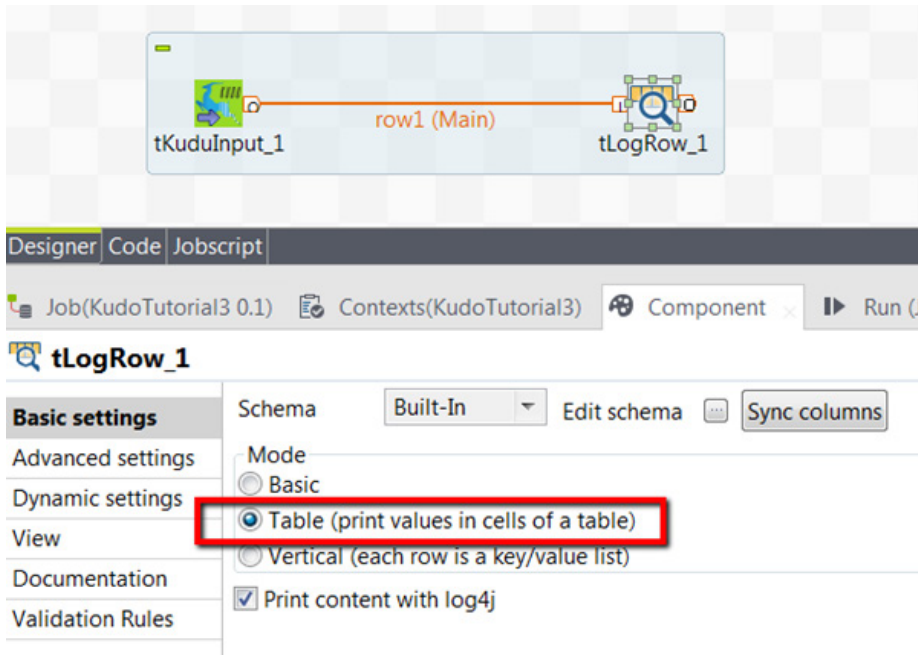


7. We select the tLogRow component from the Palette and drop it on the job view panel.

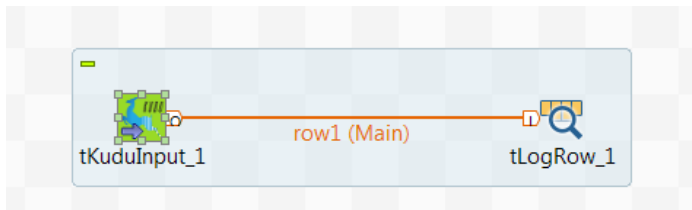




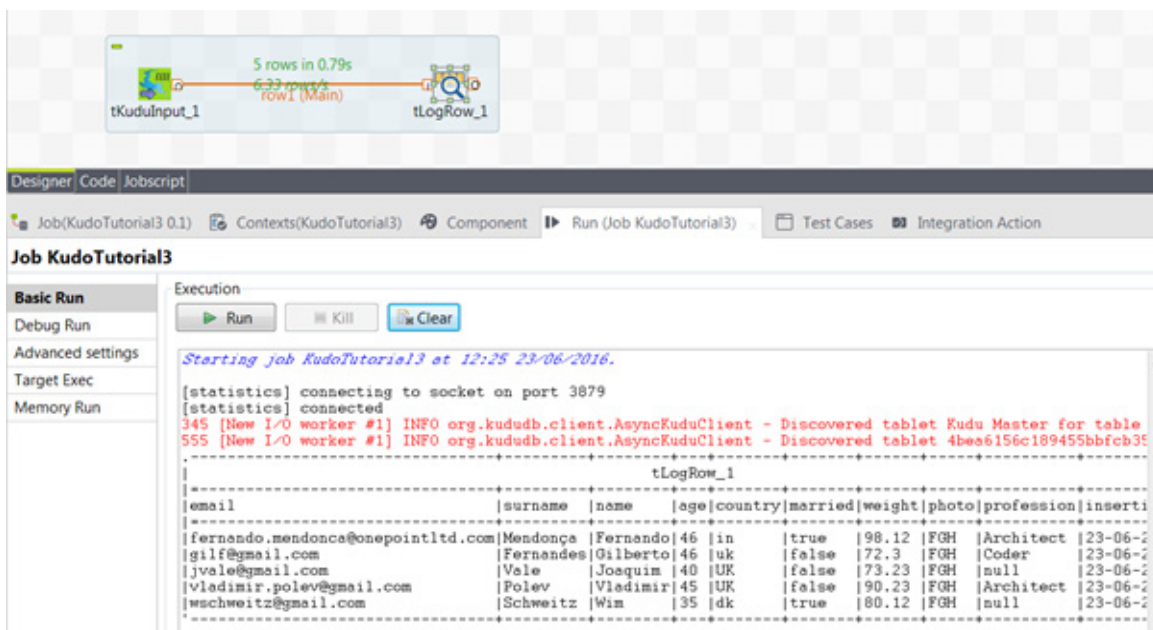
8. Now double-click on the tLogRow component and select the "Table" mode.



9. Now we create a row connection from the tKuduInput component to the tLogRow component.



10. The job can now be executed. In case of success you will see the following:



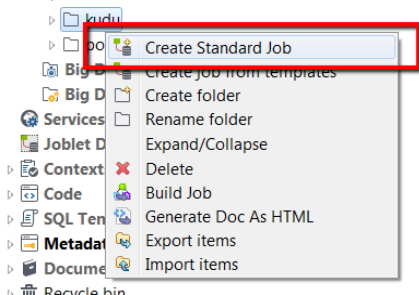
EXAMPLE JOB 2

In this example job you will learn how to setup the tKuduInput component and how to perform a user defined queries scan with the tKuduInput component.

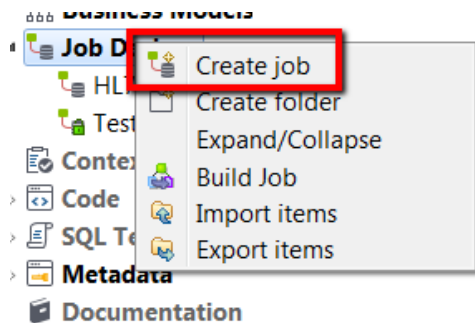
Step by step instructions

1. We will start by creating a standard Talend job (if you are using the "Enterprise version"). If you are using the open source version of Talend you just typically create a normal job.

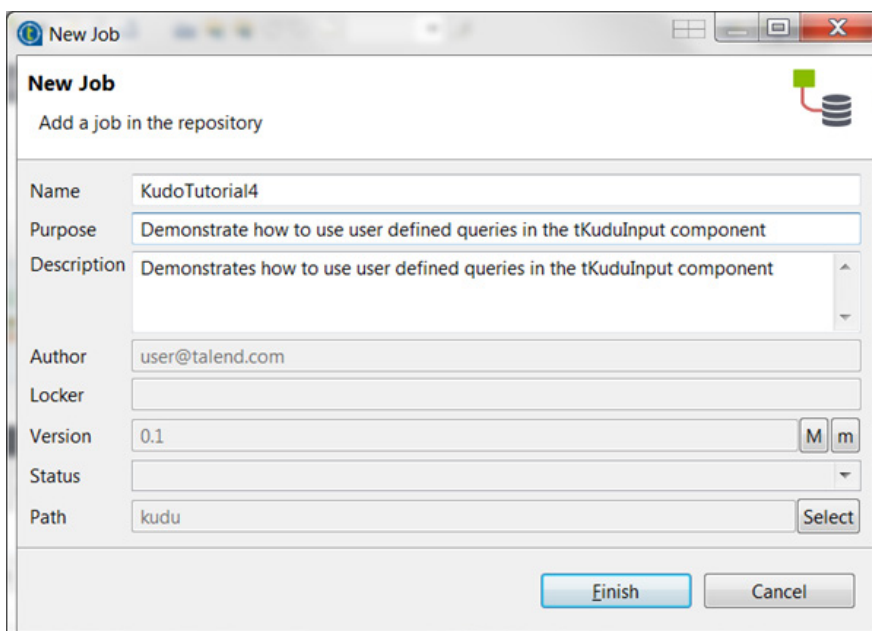
a. Enterprise version



b. TOS version



2. Here are the details of the created job.



New Job
Add a job in the repository

Name: KudoTutorial4

Purpose: Demonstrate how to use user defined queries in the tKuduInput component

Description: Demonstrates how to use user defined queries in the tKuduInput component

Author: user@talend.com

Locker:

Version: 0.1 [M] [m]

Status:

Path: kudu [Select]

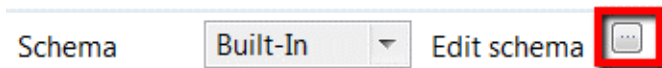
[Finish] [Cancel]



3. We select the tKuduInput component from the Palette and drop it on the job view panel.



4. We double-click on the created tKuduInput component and click on the "Edit schema" button.



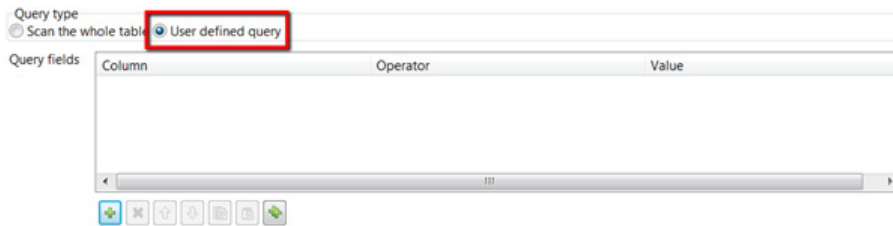
5. The schema we are going to create describes a customer. It contains the following fields:

- a. Email (the primary key)
- b. Surname
- c. Given name
- d. Age
- e. Country
- f. Married
- g. Weight
- h. Photo
- i. Profession
- j. Insertion Date

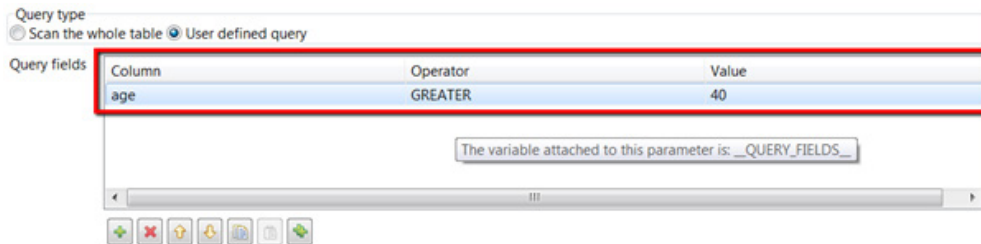
| Column | Db Column | K. | Type | N. | Date Pattern (Ctrl+Sp... | Length | Precision | Default | Comment |
|---------------|---------------|-------------------------------------|---------|-------------------------------------|--------------------------|--------|-----------|-------------|---------|
| email | email | <input checked="" type="checkbox"/> | String | <input type="checkbox"/> | | | | | |
| surname | surname | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | | |
| name | name | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | | |
| age | age | <input type="checkbox"/> | Integer | <input checked="" type="checkbox"/> | | | | | |
| country | country | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | "UK" | |
| married | married | <input type="checkbox"/> | Boolean | <input checked="" type="checkbox"/> | | | | | |
| weight | weight | <input type="checkbox"/> | Double | <input checked="" type="checkbox"/> | | | | | |
| photo | photo | <input type="checkbox"/> | byte[] | <input checked="" type="checkbox"/> | | | | | |
| profession | profession | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | | | | |
| insertionDate | insertionDate | <input type="checkbox"/> | Date | <input checked="" type="checkbox"/> | "dd-MM-yyyy" | | | new java... | |

Hint: if you have already gone through the previous example jobs (Example Job 1, Example Job 2) you can simply copy the schema from the tFixedInputFlow component. Or you can import simply the schema file provided in this tutorial (see chapter Support Materials).

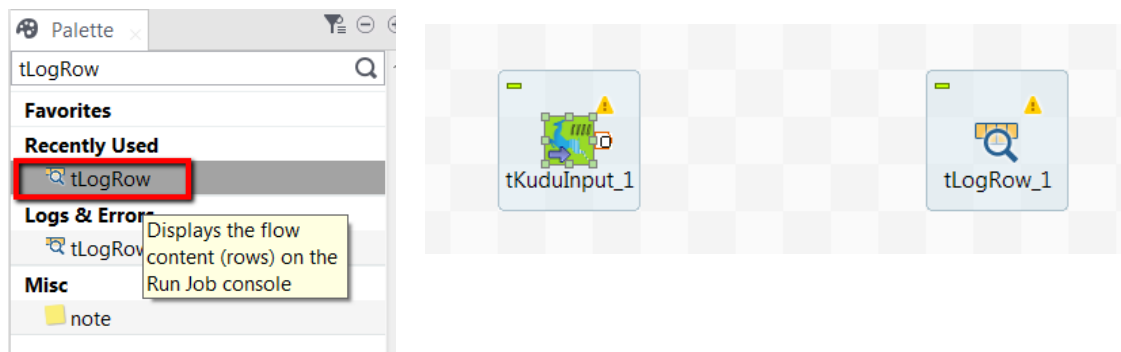
6. We are going first to create a query which filters out all customers which are older than 40. In order to create such a query, double click on the tKuduInput component and select "User defined query"



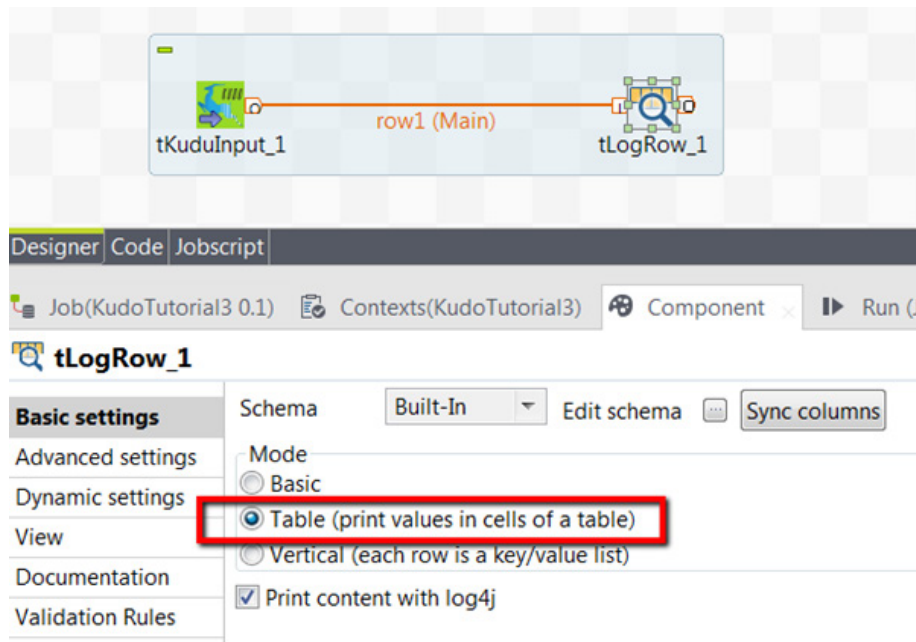
7. Now add one line to the query by pressing the “ + ” button. Select the “age” column and the “GREATER” operator. Write into the “Value” field “40” (with no quotes).



8. We select the tLogRow component from the Palette and drop it on the job view panel.

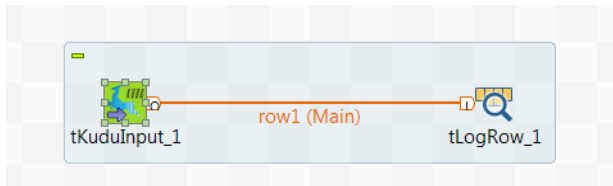


9. Now double-click on the tLogRow component and select the “Table” mode.





10. Now we create a row connection from the tKuduInput component to the tLogRow component.



11. Now you can run the job for the first time and you will see that all customers listed on the console are 40+ of age:

The screenshot shows the execution of a Talend job. The console output displays the following table:

| email | surname | name | age | country | married | weight | photo | profession |
|-----------------------------------|-----------|----------|-----|---------|---------|--------|-------|------------|
| Fernando.mendonca@onepointltd.com | Mendonça | Fernando | 46 | in | true | 98.12 | FGH | Architect |
| gilf@gmail.com | Fernandes | Gilberto | 46 | uk | false | 72.3 | FGH | Coder |
| vladimir.polev@gmail.com | Polev | Vladimir | 45 | UK | false | 90.23 | FGH | Architect |

12. Now let us change the existing filter and try to find a user by email address. Double click on the tKuduInput component and remove the existing filter and add the following filter:

The screenshot shows the filter configuration in the tKuduInput component. The filter is set to 'email' with the operator 'EQUALS' and the value 'gilf@gmail.com'.

| Column | Operator | Value |
|--------|----------|------------------|
| email | EQUALS | "gilf@gmail.com" |

13. Now run the job again and you will see that there is only one single entry in the output.

The screenshot shows the execution of the Talend job after the filter change. The console output displays the following table:

| email | surname | name | age | country | married | weight | photo | profession | insertionDate |
|----------------|-----------|----------|-----|---------|---------|--------|-------|------------|---------------|
| gilf@gmail.com | Fernandes | Gilberto | 46 | uk | false | 72.3 | FGH | Coder | 23-06-2016 |



14. Let us now create a combined filter which filters by age and by country. Add the following lines to the query fields:

| Query fields | Column | Operator | Value |
|--------------|---------|----------|-------|
| | age | GREATER | 40 |
| | country | EQUALS | "in" |

15. Now run the job again and you will see all customers which are associated to "in" and over 40.

Jobscript

orial4 0.1 Contexts(KudoTutorial4) Component Run (Job KudoTutorial4) Test Cases Integration Action

orial4

Execution

Run Kill Clear

Starting job KudoTutorial14 at 13:18 23/06/2016.

```
[statistics] connecting to socket on port 3990
[statistics] connected
341 [New I/O worker #1] INFO org.kududb.client.AsyncKuduClient - Discovered tablet Kudu Master for table
386 [New I/O worker #1] INFO org.kududb.client.AsyncKuduClient - Discovered tablet 4bea6156c189455bbfcb3
```

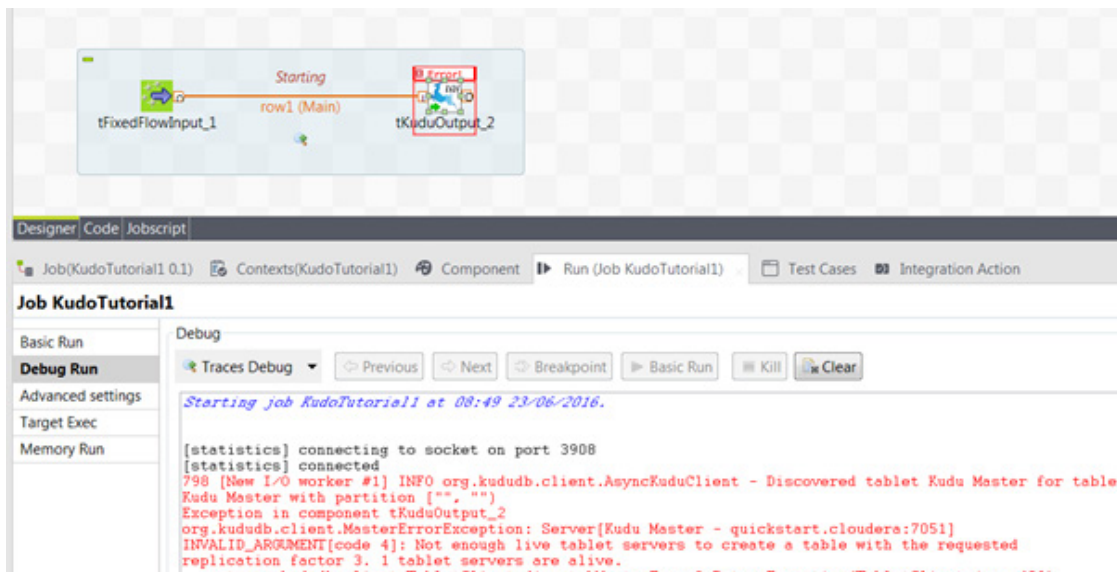
| tLogRow_1 | | | | | | | | | |
|-----------------------------------|----------|----------|-----|---------|---------|--------|-------|------------|----------|
| email | surname | name | age | country | married | weight | photo | profession | inserti |
| fernando.mendonca@onepointltd.com | Mendonça | Fernando | 46 | in | true | 98.12 | FGH | Architect | 23-06-20 |

Common Errors

REQUESTED REPLICATION FACTOR

One of the most common errors that you will probably get when you run the job for the first time on a test environment using a single virtual machine is:

"org.kududb.client.MasterErrorException: Server[Kudu Master - quickstart.cloudera:7051] INVALID_ARGUMENT[code 4]: Not enough live tablet servers to create a table with the requested replication factor 3. 1 tablet servers are alive."

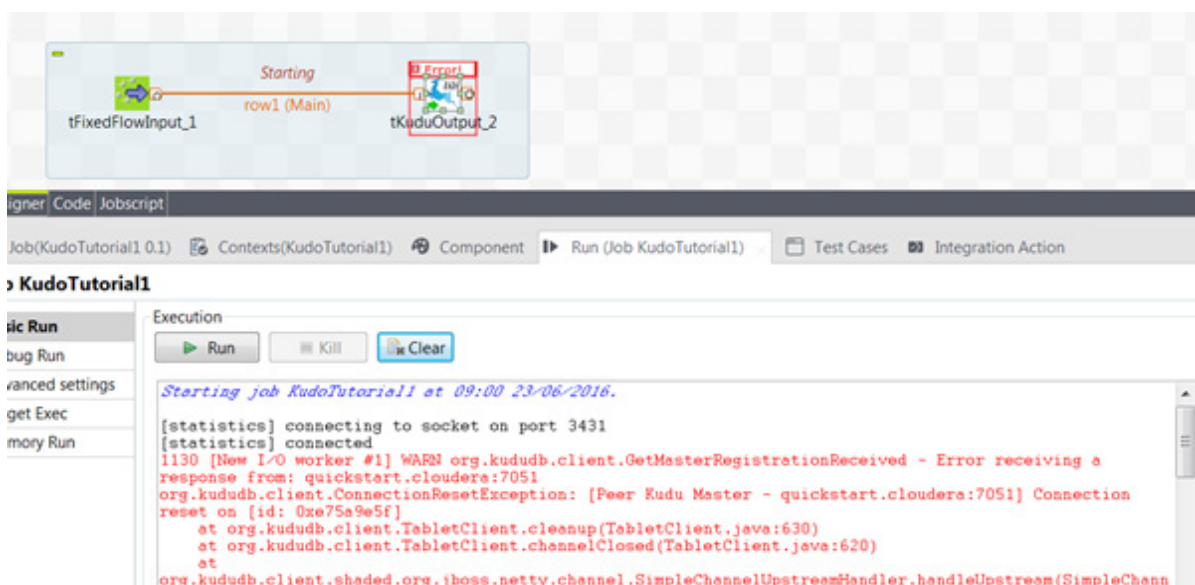


SOLUTION

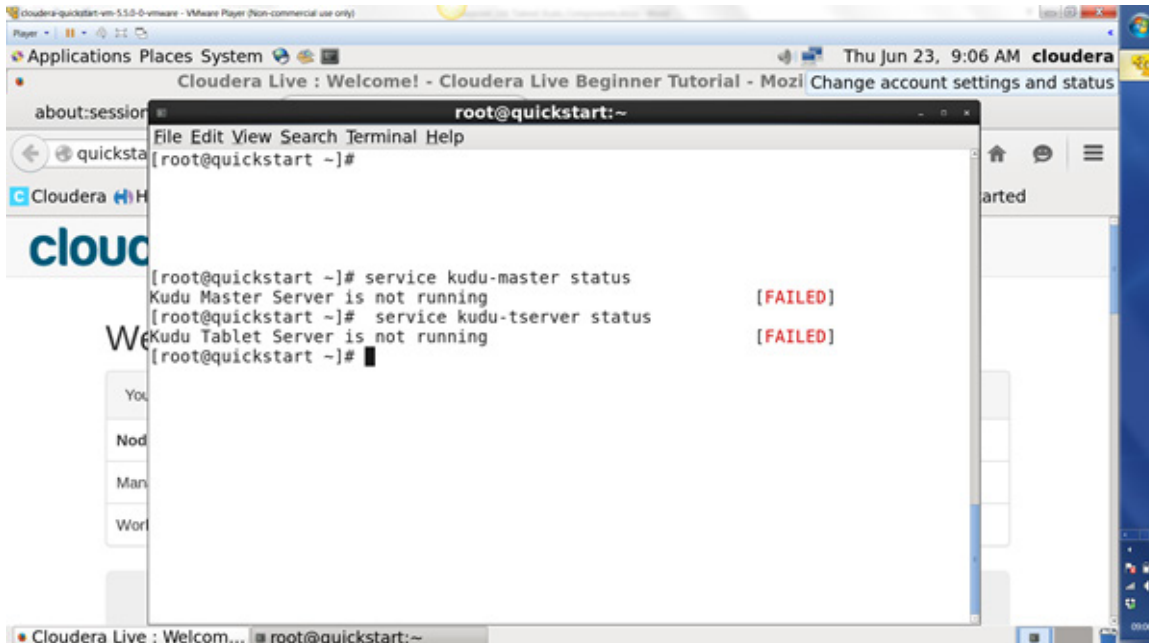
Simply set the requested replication number in the **Advanced Settings** tab to 1 in this case:

Connection Failure

This problem occurs when the Kudu services have not been started properly. Typically this is what you see on your screen:



Typically there is nothing you can do in Talend about this. You should in this case check, if the two Apache Kudu services are running on the Apache Kudu server:



```
root@quickstart:~# service kudu-master status
Kudu Master Server is not running [FAILED]
root@quickstart ~]# service kudu-tserver status
Kudu Tablet Server is not running [FAILED]
root@quickstart ~]#
```

We suggest in this case to try to start the services with:

```
service kudu-master start
service kudu-tserver start
```

More information about Kudu administration can be found on this page:

<http://getkudu.io/docs/troubleshooting.html>

