



How to Guide:

# Connect Tableau to ArangoDB

*In this tutorial we will use Tableau Desktop with the Tableau Web Data Connector for ArangoDB using ArangoDB 3.4.*

---

*The Foxx service is not an official Tableau connector, it uses the Web Data Connector and could be customized to allow fit for individual needs. Tableau is a trademark of Tableau Software, Inc.*

In this tutorial we will use Tableau Desktop with the Tableau Web Data Connector and ArangoDB 3.4.

This tutorial assumes you have access to a running instance of ArangoDB that already contains the data you want to export to Tableau.

## Installing the connector

The Tableau connector can be installed as a Foxx service using the ArangoDB web interface or the Foxx CLI.

To install the service using the web interface:

1. Open the ArangoDB web interface in your browser (e.g. `http://localhost:8529` if ArangoDB is running locally).
2. Enter your ArangoDB credentials and select the database from which you want to export data to Tableau.
3. Select the *Services* tab on the right, press *Add Service* and select the *tableau-connector* service.
4. Enter a mount point (e.g. `/tableau`) and press *Install*.

To install the service using the Foxx CLI use the following command (assuming user `root`, database `_system` and mount point `/tableau`):

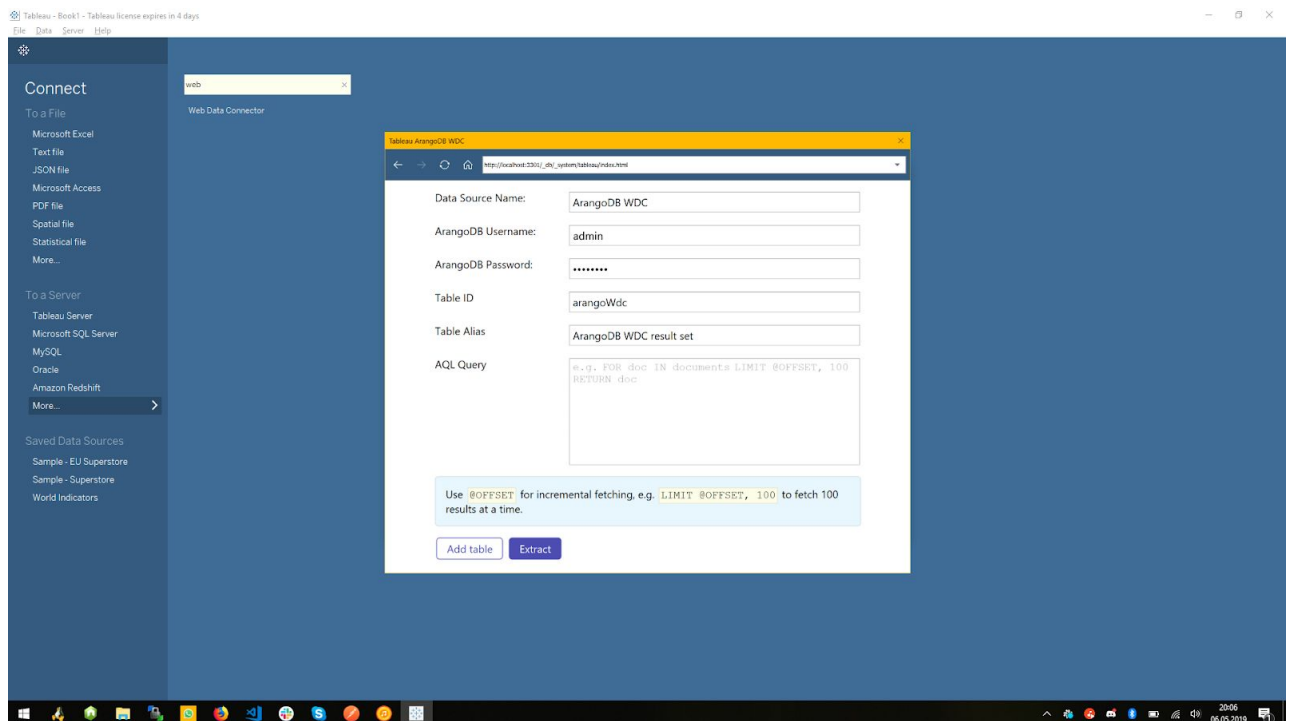
```
foxx install -u root -P -H http://localhost:8529 -D _system /tableau \
https://github.com/arangodb-foxx/tableau-connector/archive/master.zip
```

## Using the connector in Tableau Desktop

We'll start out by connecting a new data source from the *Connect* sidebar on the home screen by navigating to *To a server > More...* and selecting the *Web Data Connector* data source. You can also find the connector more easily by entering "Web" in the search box.

If you are using an existing Tableau workbook, you can also access this dialog by pressing the *New Data Source* icon in the toolbar.

In the *Web Data Connector* dialog enter the URL of the Tableau Web Data Connector service. For example, if we have a local copy of ArangoDB running on our machine and the Tableau connector was installed at `/tableau`, our URL looks like this: `http://localhost:8529/_db/_system/tableau`



The **Data Source Name** is the name Tableau will use to identify the data source in the workbook's list of data sources. There is no need to change this value if the default is unambiguous to you.

The **ArangoDB Username** and **ArangoDB Password** must be valid ArangoDB credentials that grant at least read access on all collections you want to load data from. It is recommended that you let your administrator create a dedicated ArangoDB user account that only holds the minimum permissions Tableau needs to access the data.

The **Table ID** and **Table Alias** are the machine readable and human readable names Tableau will use to identify the query in the workbook. If you only use a single query there is no need to change these values except for clarity.

The **AQL Query** must be a well-formed AQL query that will generate data for Tableau. For example the following query returns all documents in the collection `data`:

```
FOR doc IN data
RETURN doc
```

When reading from large collections you should use pagination using AQL's built-in `LIMIT` operation and the `@OFFSET` bind variable provided by the connector. The following example limits results to 200 documents per request:

```
FOR doc IN data
LIMIT @OFFSET, 200
RETURN doc
```

Note that Tableau expects flat data and may not always know how to handle nested objects or arrays. You can use the AQL functions `KEEP` and `UNSET` to omit problematic values:

```
FOR doc IN data
LIMIT @OFFSET, 200
RETURN UNSET(doc, ["nestedObject", "someArray"])
```

Generally the Tableau connector supports three types of return values:

- An object mapping field names to scalar values, in this case the field names will be used as column names.

Example (columns "field1", "field2"):

```
LET a = "something scalar"
LET b = 1234
RETURN {field1: a, field2: b}
```

- An array of scalar values, in this case column names will be generated.

Example (columns "value1", "value2", "value3"):

```
LET a = "something scalar"
LET b = 1234
LET c = true
RETURN [a, b, c]
```

- Any scalar value (e.g. a string, number or boolean), in this case the value will be treated as a single column

Example (single column "value"):

```
LET value = "something scalar"  
RETURN value
```

Note that the Tableau connector will always add an extra whole number column with the ID `_i` for technical reasons that appears in Tableau with the human readable alias name "Index".

To export results from additional queries, press the *Add table* button to expand the form with additional query fields. Unused queries can be removed by pressing the *Remove* icon next to the top right of the query fieldset.

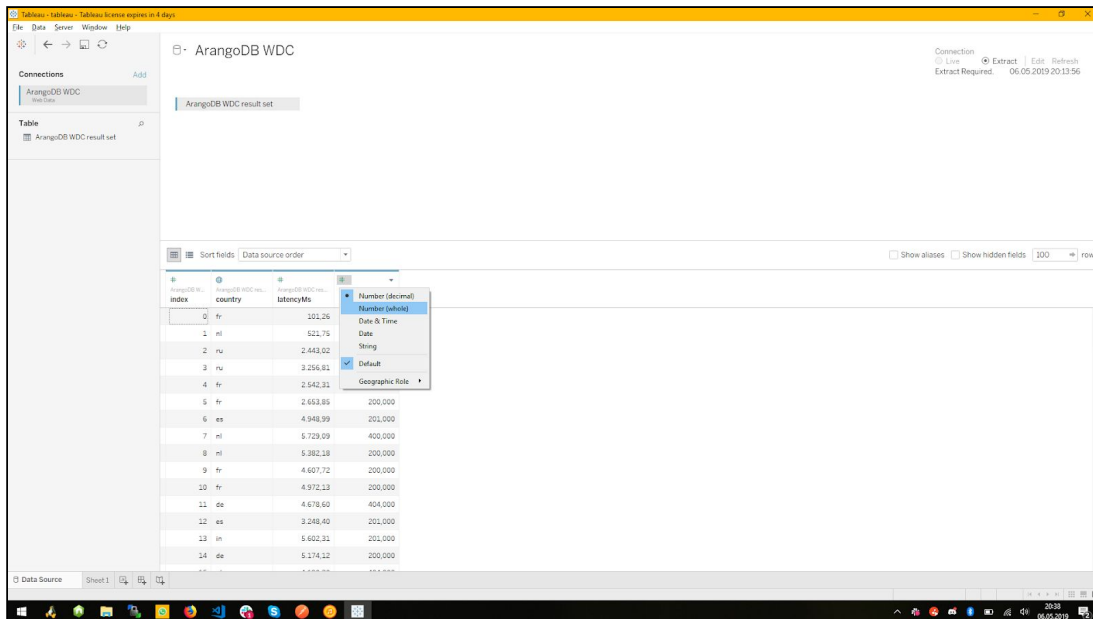
Remember that you can try out AQL queries in the ArangoDB web interface before using them in Tableau. This can be useful when troubleshooting import errors because of data formats Tableau does not understand.

Once you're satisfied with your queries, press the *Extract* button to proceed to the next step.

## Data selection

If the connection succeeded, you should now see an empty table. Press the *Update Now* button to see the first page of results from your query.

The Tableau connector tries to infer appropriate types for the query results but depending on your data you may want to adjust these types.



You can change the type of each field by pressing the little icon in the top left of the column. You can also rename the fields by clicking the little triangle icon that appears in the top right when hovering over the column label.

While it is also possible to remove columns or create derived values at this point, it is more efficient to adjust the AQL query to perform these changes outside Tableau.

Once you're satisfied with your changes, click on the worksheet in the bottom bar to proceed.

## Creating a worksheet

With all the data imported into Tableau, you can now visualize it by dragging fields onto the worksheet.

