Umongo is a one of the GUIs to work with MongoDB

Latest version can be found here: <http://edgytech.com/umongo/>

Also , there are lot of analogs , other GUIs can be found here: <https://docs.mongodb.org/ecosystem/tools/administration-interfaces/>

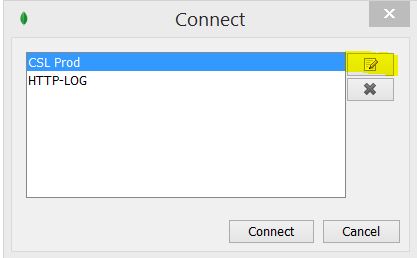
Basically most of them uses the same syntax for querying – it’s JSON-style querying, which is based on MongoDB shell

Servers to connect to:

1. Dev server: http-log.corp.ooma.com , port number is by default (27017)
2. Prod server: mobilelogs.corp.ooma.com , the same default port(27017)

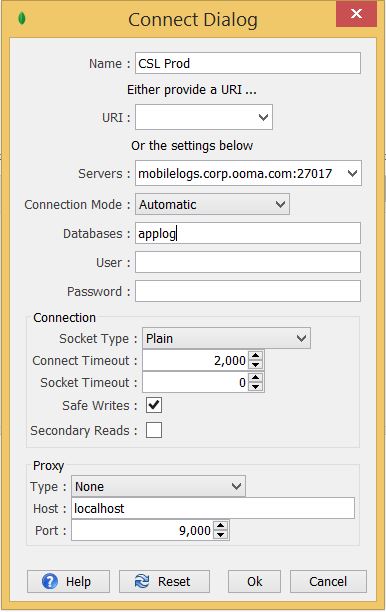
After launch of Umongo application click File(in the top left corner) - Connect

After click next window will appear:



This yellow marked button –to create new or to modify existing connection

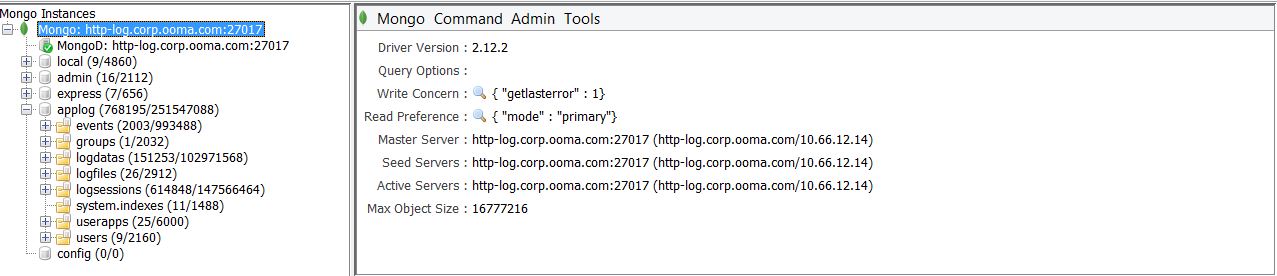
Here is a sample of connect dialog window:



\*Hint: not necessary to set Database( applog in example) because during connect process to MongoDB instance switch between databases function is available

Press OK , then press Connect

If all fields are filled with proper values , then a list of databases is shown , every database could be expanded by clicking “+”



In this sample “applog” – is a name of DB ;

Events, groups, logdatas etc. – collections(the same as tables in SQL-like databases)

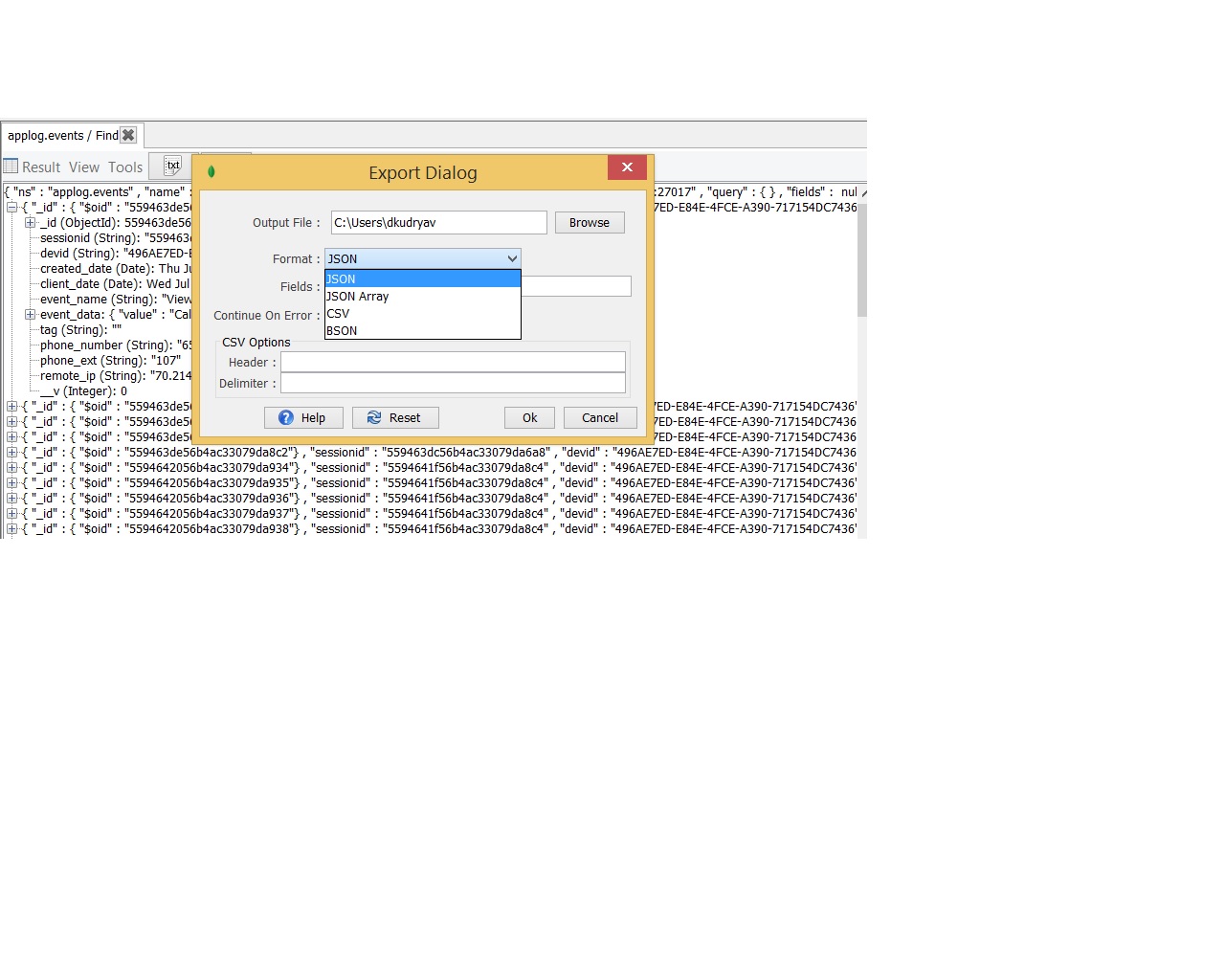
Any collection can be chosen by left mouse button and its statistics will be displayed on a right side of view – number of records, size of this collection , average size of one document(the same as row in SQL-like DB) etc.

Actions menu is available by clicking right mouse button.

Also, same functionality can be found in menu, which is on the top of window

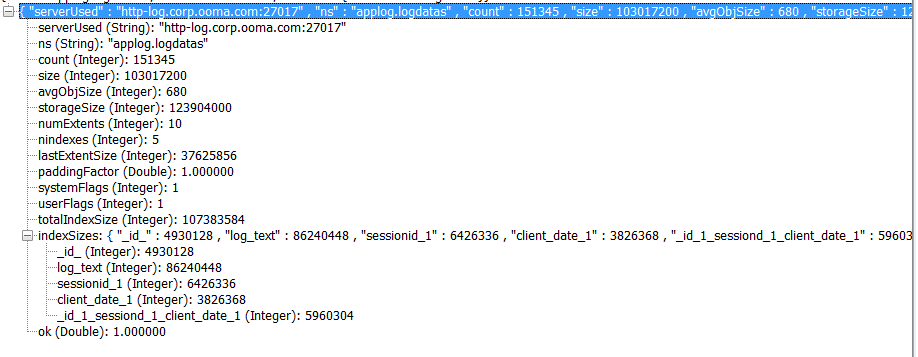
This menu contains:

1. Collections
2. Create index
3. Refresh
4. Rename
5. Settings
6. Import File -> to insert documents from file to this collection
7. Export to File -> to export data from mongo to file using following formats:



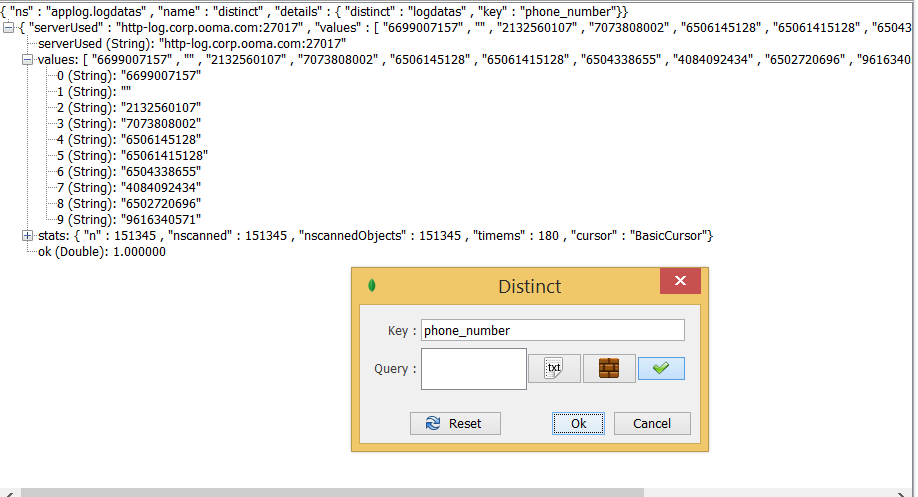
1. Drop collection
2. Command
3. GetStats – to generate statistics about this collection

Example:



1. Get Indexes – to get all indexes for this collection
2. Distinct

To get list of unique data in collection with pre-filter functionality enabled (before getting unique fields)



Key – field name for searching unique data

Query – add a filter if needed ( e.g. if only unique phone numbers added in current month needed)

1. Aggregate – MongoDB Aggregation framework , mostly used for executing report-like queries

Additional info can be found here <https://docs.mongodb.org/manual/core/aggregation-introduction/>

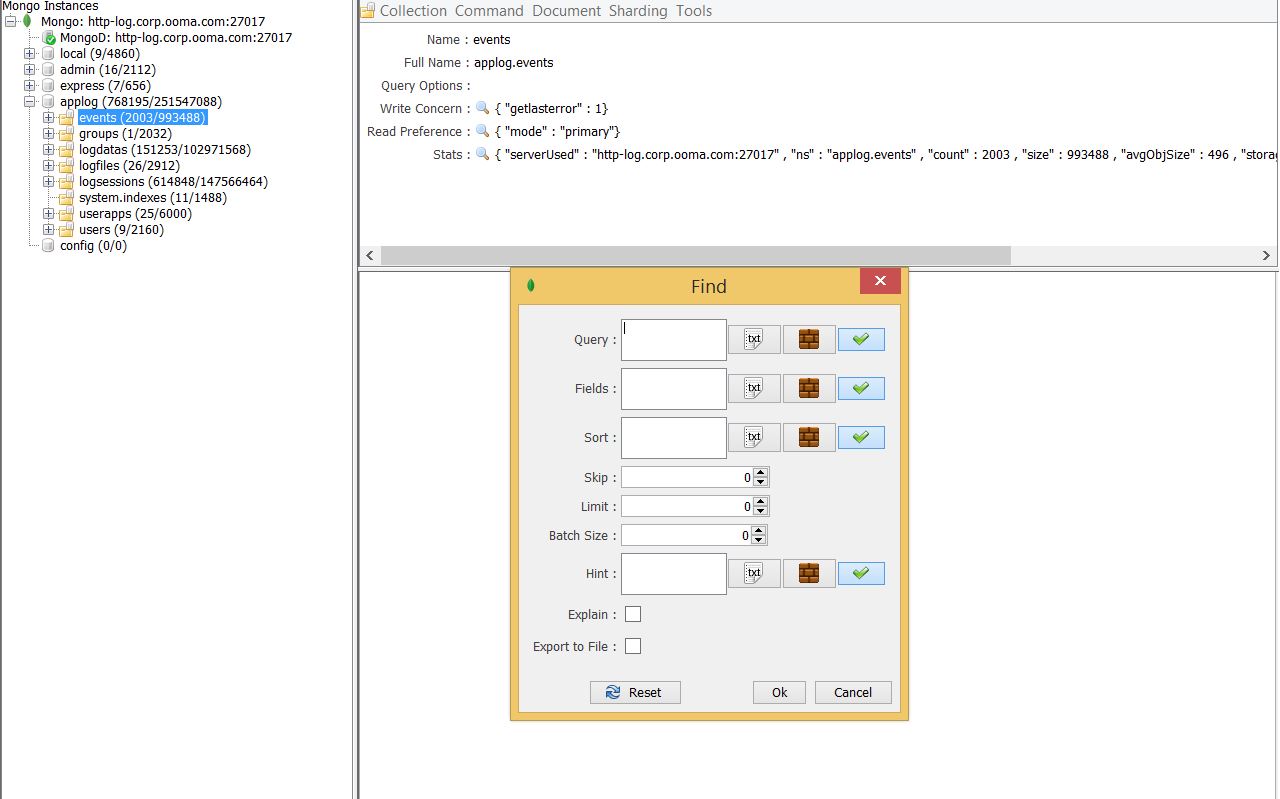
1. Map reduce – to work with large files,

document from mongodb.org about this feature:

<https://docs.mongodb.org/manual/core/map-reduce/>

1. Geo Near
2. Document
3. Find

So, this is main feature which will be used



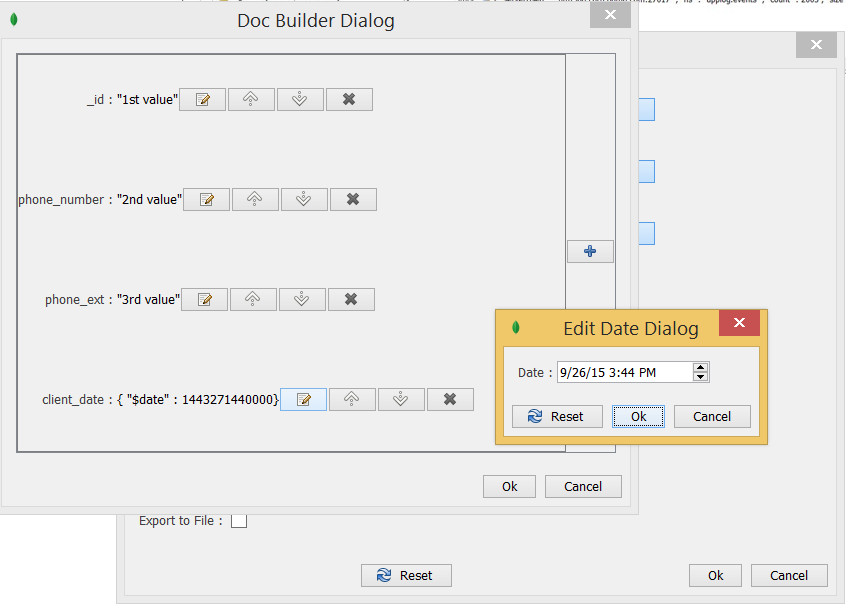
There are following fields in Find menu:

1. **Query** – analogue of sql-like **where** part

Is used for sorting data in collection, this field can be filled manually using JSON syntax:

{"\_id":"1st value", "phone\_number" : "2nd value" , "phone\_ext" : "3rd value" }

or using query builder (brown button on the right of this field)

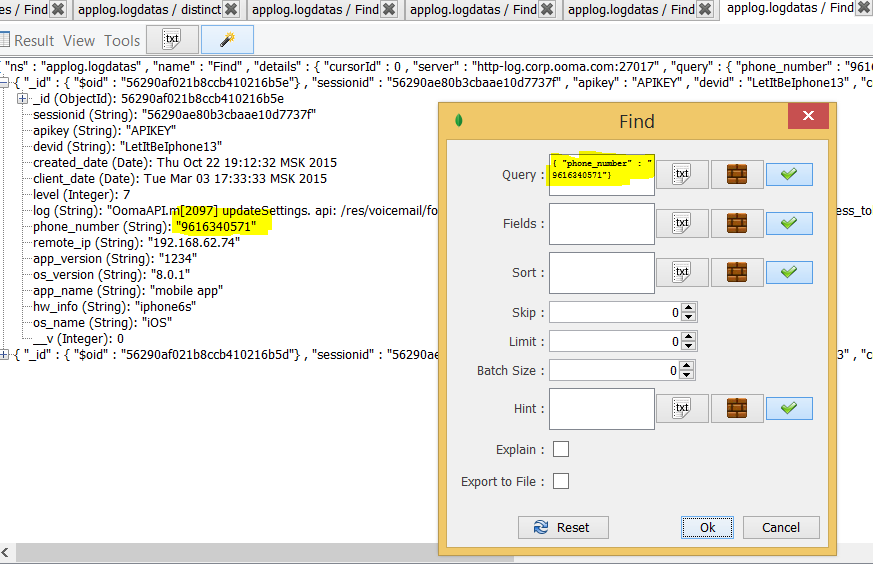


In Builder Dialog field name must be added from collection (to look at field’s list run FIND with empty field, first 100 docs from collection will be displayed)

For every field name data type must be chosen, all fields(except date fields – client\_date and creation\_date) in our DB are “String” type.

If day-like filter is needed select DATE type. After adding a field set its value using button Edit Field near every added field. For DATE Type there will be a calendar(shown above)

After setting all needed fields press ok button and wait for UMongo to fetch results.



In this example only 2 documents were fetched based on filter.

1. **Fields** – in SQL-like syntax it is a part of **SELECT** <what fields to show>

Filling this field is the same as for **Query** –

1. JSON-like

**{"phone\_number" : 1}**

0 – hide

1 – show

By default if something is entered here all fields will be hidden , **except** fields with “1” value and MongoDB “\_id”, if “\_id” field is not needed manually switch off “\_id” field , our JSON string will be like this

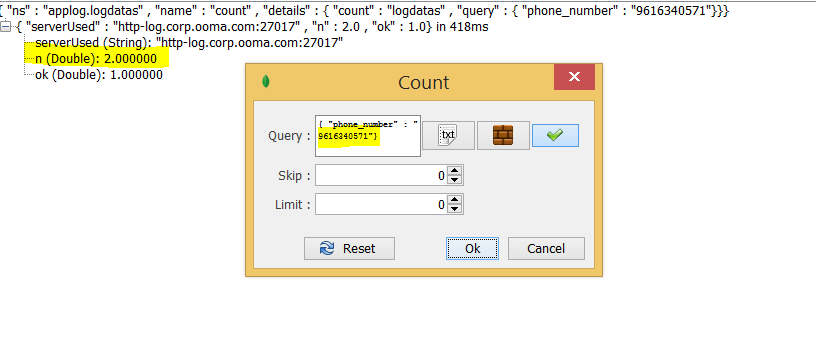
**{"phone\_number" : 1,"\_id":0}**

1. Query builder – the same as for queries, but here only 2 values can be set : 1 and 0 ( show and hide)
2. **Sort** – in SQL-like syntax it is a part of **ORDER BY**
3. Json-like

**{"phone\_number" : 1}**

1. Query builder – the same as for queries, but only 2 values can be set : 1 and 0 ( show and hide)
2. **Skip** – how many documents to skip before data-fetch
3. **Limit** – how many documents to fetch
4. **Count** – in SQL-like syntax analogue – **count()** function

Here is an example:



There are only 2 documents in a collection with phone\_number 9616340571

1. Insert – to add a document into collection
2. Update – to update a document(s) in this collection using some rules
3. Remove – to delete document from collection

4 ) Sharing – MongoDB features based on shared cluster ,all data can be spread between servers using shared key (for example phone\_number) .

5 ) Tools