Query Designer

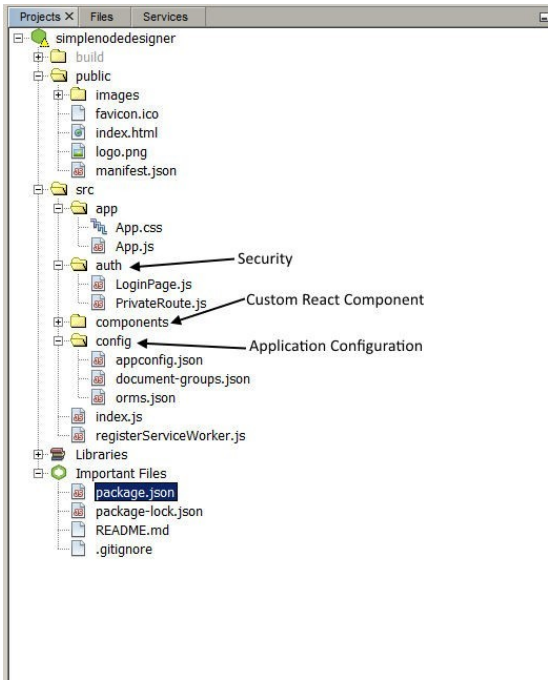
Rob Tucker
04/11/2019

Query Designer

Contents

The Query Designer is a React web application designed to work in concert with Simple Node ORM installations. The query designer allows users to create query documents that can return JSON result sets or JSON object graphs. Once created, the query documents can be accssed and run via REST calls to retrieve JSON data or as the foundation for reports built by the Report Designer..

Project Layout



The project is available on github at:

https://rbtucker@github.com/rbtucker/simplenodedesigner.git

Application Configuration

Application configuration is handled in 3 files:

appconfig.json

```
{
    "textmsg": {
        "logintitletext": "Simple Node ORM",
        "modelselectdefault": "Select starting model...",
        "adddocument": "Add Document",
        "setupmenuname": "Setup",
        "filemenuname": "File",
        "newmenuname": "New",
        "newdocument": "New",
        "preferencesmenuname": "Preferences",
        "selectdata": "Select Data",
        "formatselections": "Column Settings",
        "definefilter": "Define Filter",
        "designreport": "Design Report",
        "runquery": "SQL/Results",
        "aggfunctionlabel": "Function:",
        "sortposlabel": "Sort Pos:",
        "ascdesclabel": "Desc:",
        "customcolinputlabel": "Custom:",
        "columnlabel": "Label:",
```

```
        "value": "Value:",
        "paramentrytitle": "Search Parameters",
        "savedocumenttitle": "Save Document",
        "authenticatorlabel": "Authenticator:",
        "resultformatlabel": "Result Format:",
        "validitycheckonly": "Validity Check Only",
        "distinct": "Distinct",
        "documentnamelabel": "Document Name:"
    },
    "defaultDesignAuthenticator": "DefaultAuthorizer"
}
```

name	description
textmsg	Configurable display text
defaultDesignAuthenticator	This is the authorizer to use for designer login authentication. The DefaultAuthorizer expect a basic auth string but does not validate the contents so

document-groups.json

This is the document group hierarchy used for loading/saving query documents that displays as a tree in the left pane of the main page. The “key” entry must be unique. When a document is saved it will be associated with the selected group. Document storage location is set in the ORM configuration. Documents will be stored in folders by group under this path.

```
{
    "title": "Queries",
    "key" : "grp0",
    "isLeaf": false,
    "children": [
        {
            "title": "General",
            "key" : "grp1",
            "isLeaf": false
        },
        {
            "title": "Financial",
            "key" : "grp2",
            "isLeaf": false,
            "children": [
                {
                    "title": "Accounting",
                    "key" : "grp3",
                    "isLeaf": false
                },
                {
                    "title": "Purchasing",
                    "key" : "grp4",
                    "isLeaf": false
                }
            ]

        },
        {
            "title": "Personnel",
            "key" : "grp5",
            "isLeaf": false
        }

    ]
}
```

orms.json

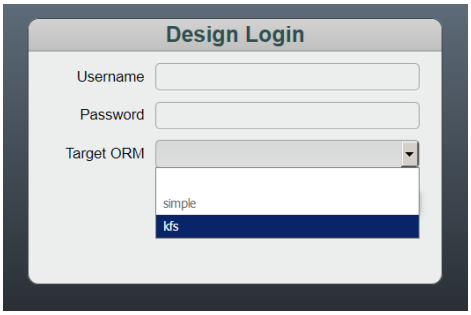
Query Designer supports multiple ORM installations. An example configuration for available ORMs is shown below. defaultUsername and defaultPassword are optional, if they exists they will auto-populate the associated login dialog fields.

```
[
{
    "name" : "simple",
    "url"  : "http://localhost:8888/orm"
},
{
    "name" : "kfs",
    "url"  : "http://localhost:8888/kfsorm",
    "defaultUsername" : "user",
    "defaultPassword" : "pass"
}
]
```

Creating a Query Document

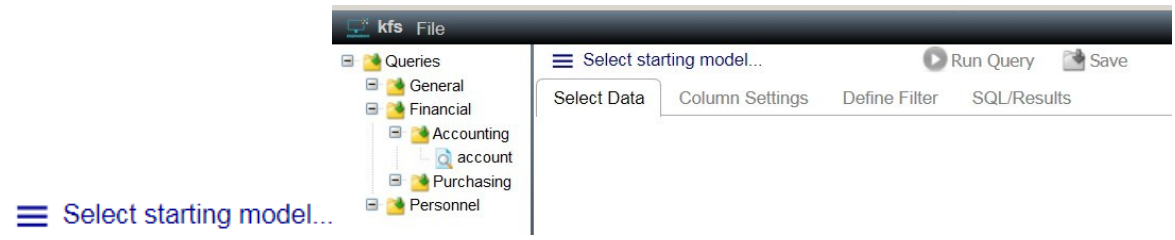
Login

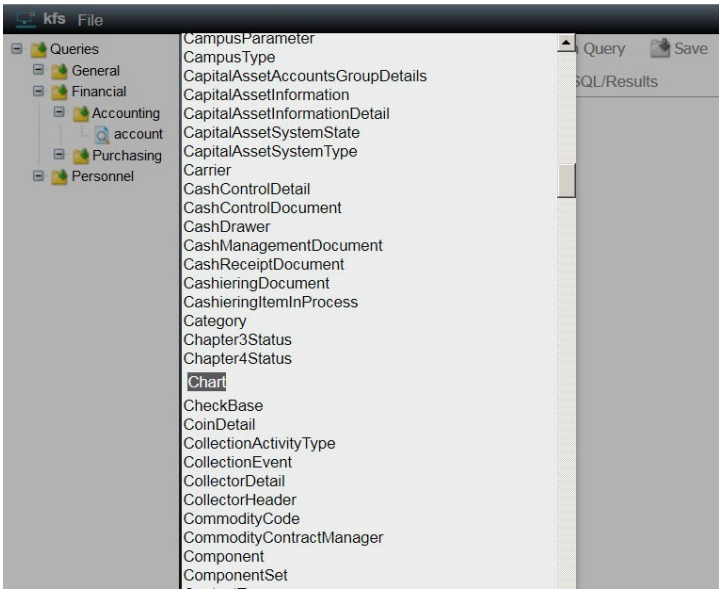
Pull up the application in the browser, select the desired ORM, enter Username and Password if required and click **Login.**



Select Starting Model

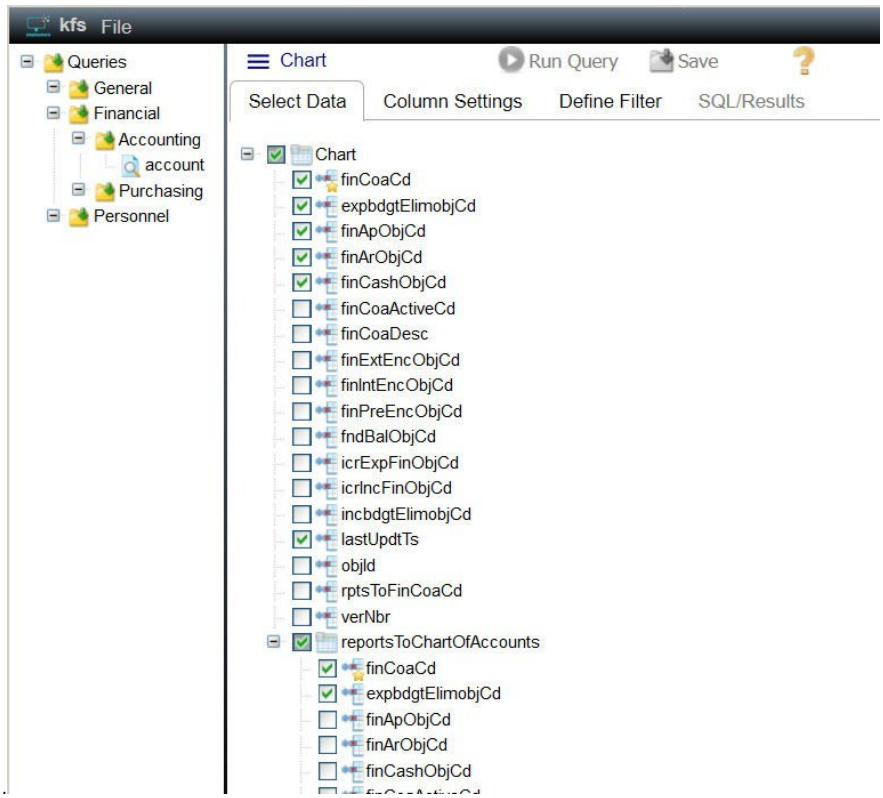
Click the menu button to display a list of available models:





Select Desired Model Columns

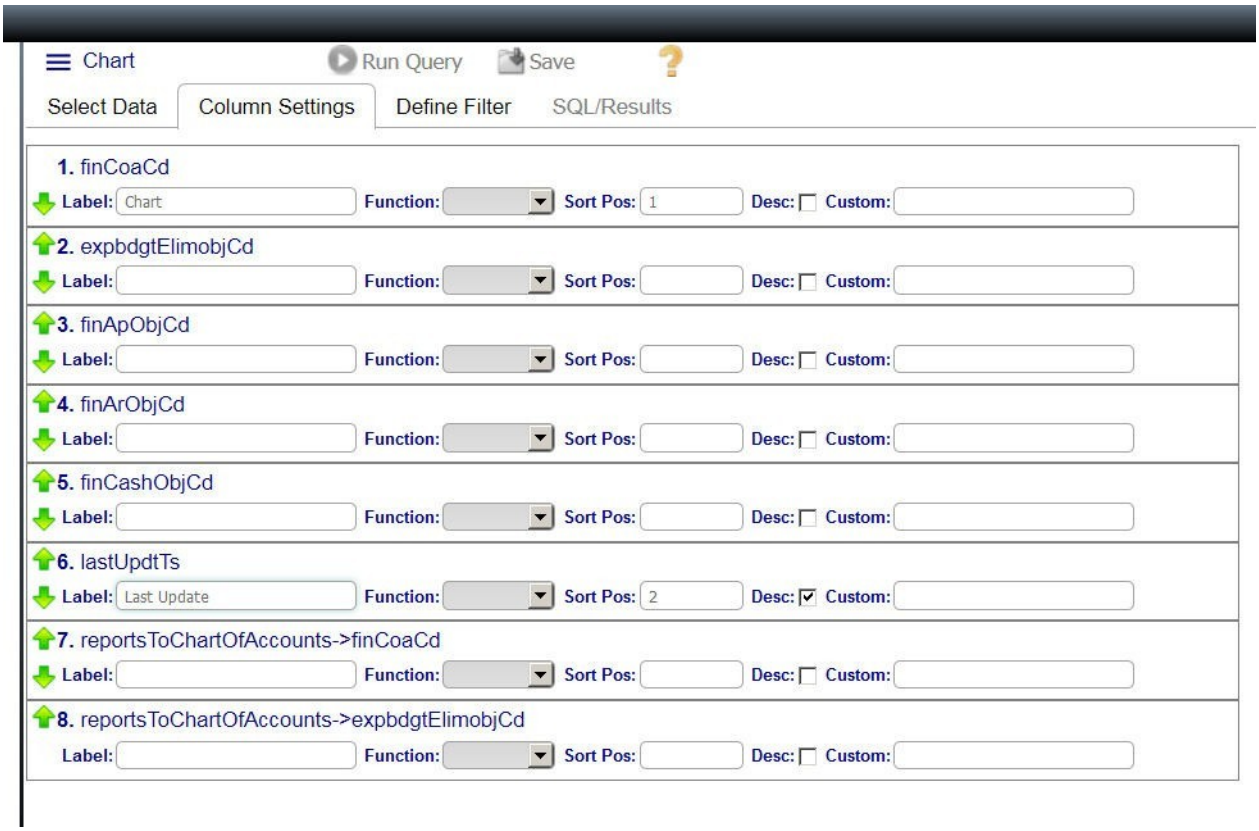
After the starting model is selected a model hierarchy will be displayed – check the desired columns. Starred columns indicate primary keys



Column Setting

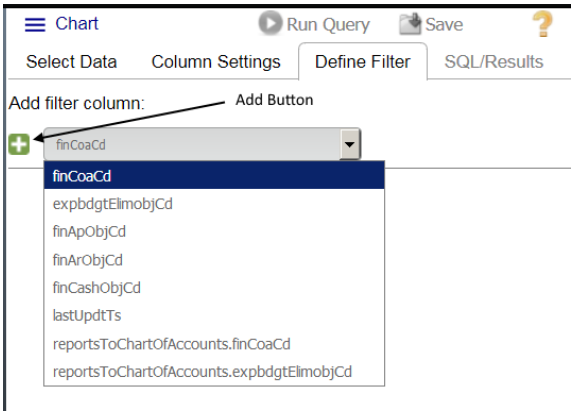
Customize the selected column settings on the Column Settings tab. The **Label** field is used in the SQL select for the “as” column setting. Aggregate functions can be applied via the **Function** dropdown. If an aggregate function is applied the appropriate “group by” clause will be generated. The **Custom** field allows the designer the flexibility to add database specific select logic. If custom is populated it will be placed in the select statement as is. Entering a “?” in the custom entry will insert the current column name into that position .

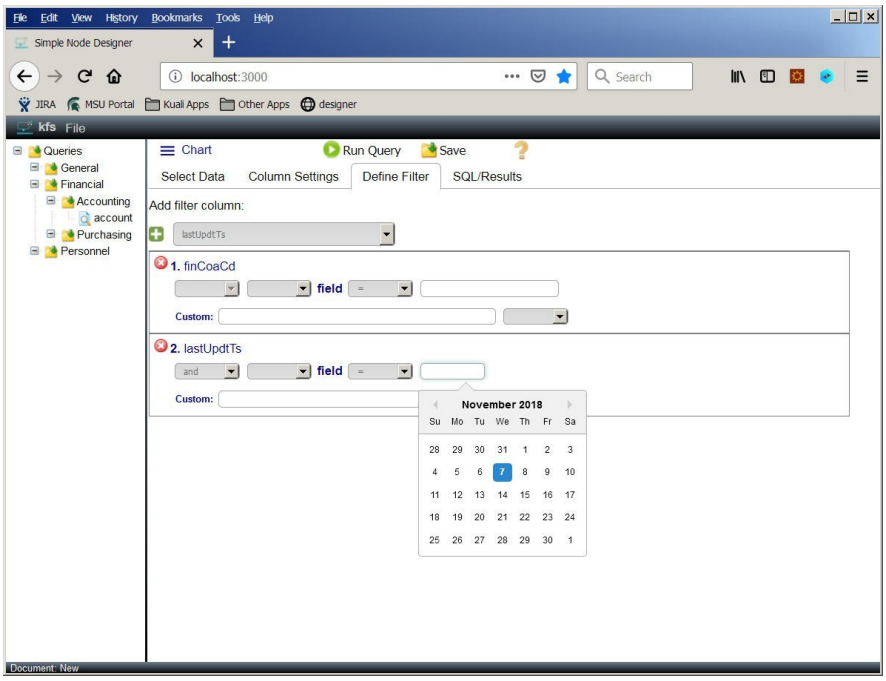
The green arrows allow you to change the select column order. If a result set is returned the columns will be in the order specified.



Define Filter

A where clause is required for all query documents. The **Define Filter** tab is to build the where clause. Where column entries are selected from the available column selections so ensure that desired filter columns are selected in the column selection tree. To add a filter entry, select the desired column and click the add button:





Each filter line allows for selection (where appropriate) of:

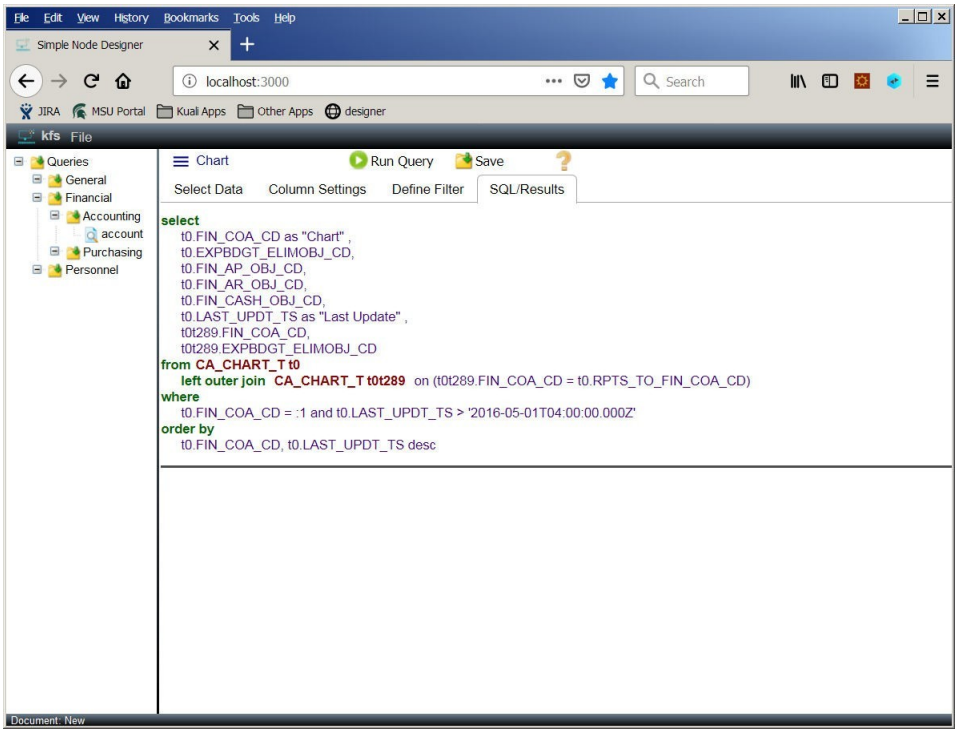
1. Logical operator (and/or)
2. Open parenthesis - (, ((, (((o ((((
3. Comparison operator (=, >, <, <=, >=, in, like, is null, is not null)
4. Comparison value entry
5. Close parenthesis -),)),))) or))))

If a comparison value field is left empty it is assumed that the field will be populated by a bind parameter when the document is run.

The **Custom** field allows for freeform entry of any where value. If this field is populated it is added to the where clause as is. Entering a “?” in the custom entry will insert the current column name into that position .

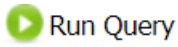
View Generated SQL

Once a filter has been defined the user can see the generated SQL in the SQL/Results tab:

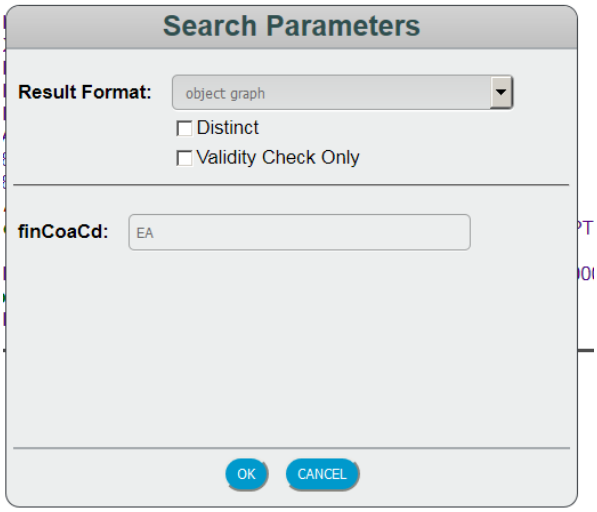


Running the Query

To run a query, click the **Run Query** button:



When **Run Query** is clicked, a parameter entry dialog will display. If any bind parameters are required they can be entered here:



The parameter entry dialog also allows the user to enter other options:

1. Result Format – object graph or result set
2. Distinct – check to run distinct query

3. Validity Check Only – if checked, no results are generated and the validity of the generated sql is displayed in the results panel.

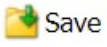
Once the required entries are made, clicking **Ok** will run the query and results will display in bottom panel of split pane:

Object Graph Result:

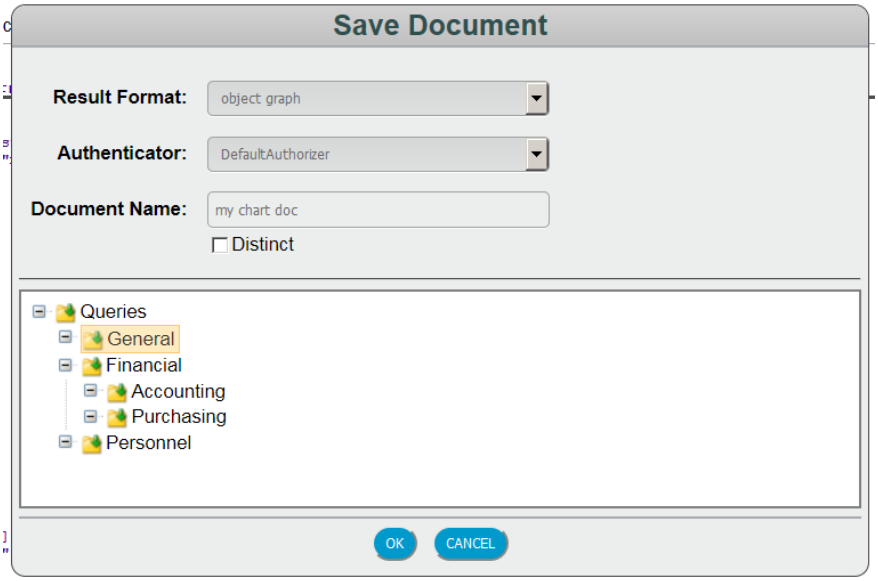
Result Set Result:

Saving the Query

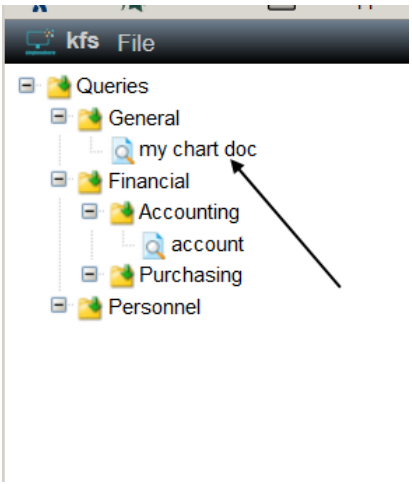
Click the save button to the Query Document



When clicked, the **Save** button will display the Save dialog:

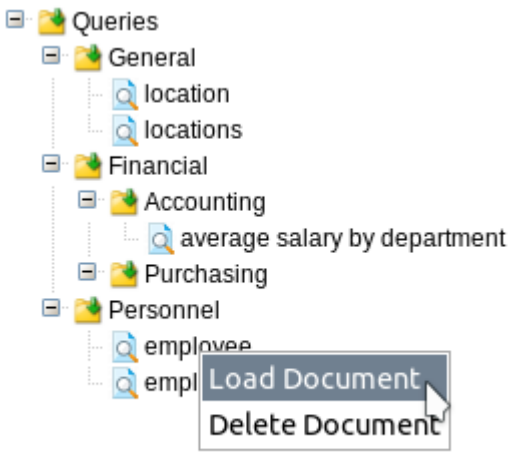


Complete the required entries and select the document group, then click **Ok** to save the document. You should see the saved document show up in the document tree in the left pane:



Loading and Deleting Existing Query Documents

To load or delete an existing query document, right click on the desired document and select the desired menu option:


Queries
General
location
locations
Financial
Accounting
average salary by department
Purchasing
Personnel
employee
empl
Load Document
Delete Document