



## Simple Node Query Designer

Rob Tucker  
11/07/2018

## Contents

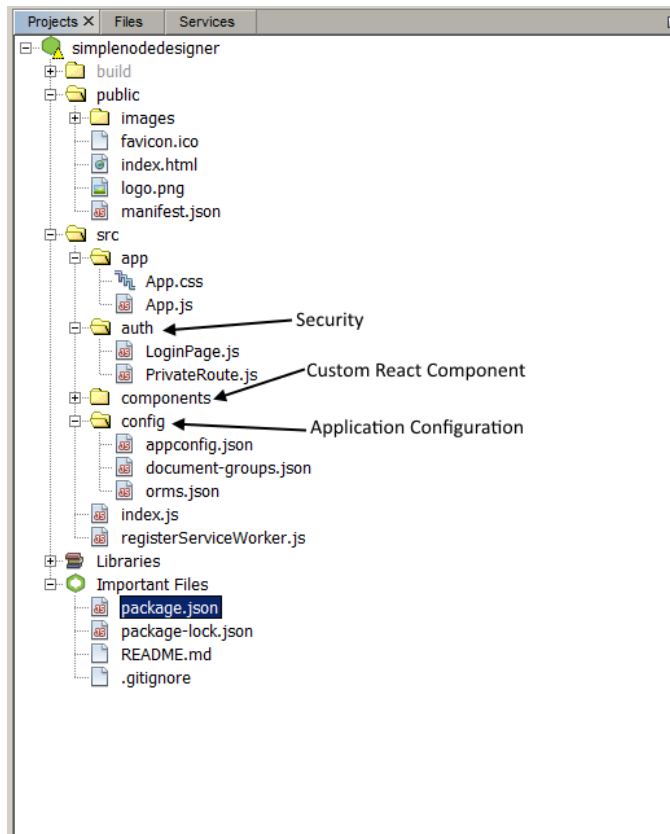
Introduction .....	2
Project Layout .....	2
Application Configuration .....	2
appconfig.json .....	2
document-groups.json .....	3
orms.json .....	4
Creating a Query Document .....	4
Login .....	4
Select Starting Model .....	4
Select Desired Model Columns .....	5
Column Setting .....	6
Define Filter .....	7
View Generated SQL .....	8
Running the Query .....	9
Saving the Query .....	10
Editing and Deleting Existing Query Documents .....	11

## Introduction

The Query Designer is a React web application designed to work in concert with Simple Node ORM installations. The query designer allows authorized users to create re-runnable query document that can return query results as JSON result sets or JSON object graphs. Once create, the query document are available via REST by document name.

## Project Layout

The Query Designer is a node React project:



The project is available in 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",
  }
}
```

## Node.js Oracle ORM for Kuali Financials

```
"modelselectdefault": "Select starting model...",
"adddocument": "Add Document",
"setupmenuname": "Setup",
"filemenuname": "File",
"newmenuname": "New",
"newdocument": "New",
"preferencesmenuname": "Preferences",
"selectdata": "Select Data",
"formatselelections": "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:"
},
"authenticators": ["DefaultAuthorizer"]
}
```

name	description
textmsg	Configurable display text
authenticators	List of available ORM JavaScript authenticator. DefaultAuthorizer is provided. Custom authenticators can be created and used. An authenticator is assigned to a saved query document to control access.

## document-groups.json

This is the document group hierarch used for saving query documents. The “key” entry must be unique. When a document is saved is will be associated with the selected group. On the ORM server the root path for document storage is set in the configuration. Document will be stored in folders by group key 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
        }
      ]
    }
  ]
}
```

## Node.js Oracle ORM for Kuali Financials

```
{
  {
    "title": "Purchasing",
    "key" : "grp4",
    "isLeaf": false
  }
},
{
  "title": "Personnel",
  "key" : "grp5",
  "isLeaf": false
}
]
```

## orms.json

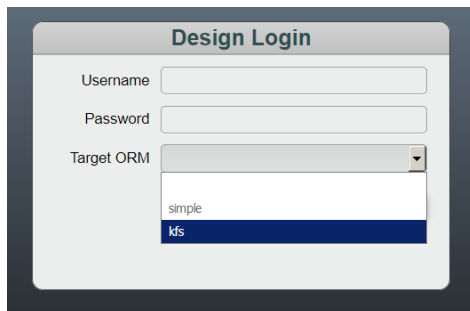
This Query Designer support multiple ORM installations. The available ORM are define as shown below. defaultUsername and defaultPassword are optional, if they exists they will auto-populate the associated login dialog fields.

```
[
{
  "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 and click **Login**.

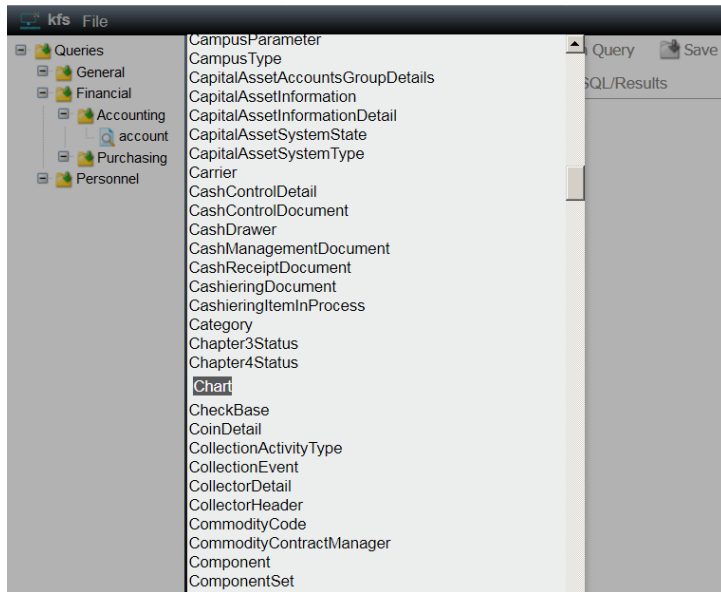
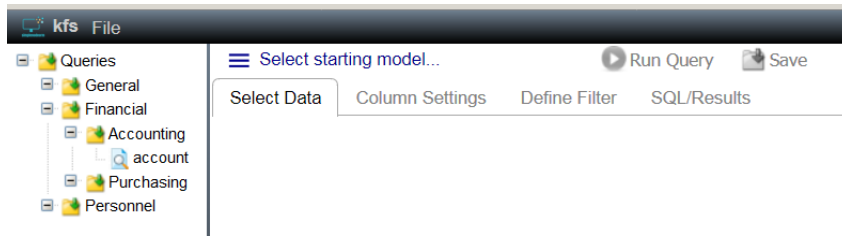


### Select Starting Model

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

 [Select starting model...](#)

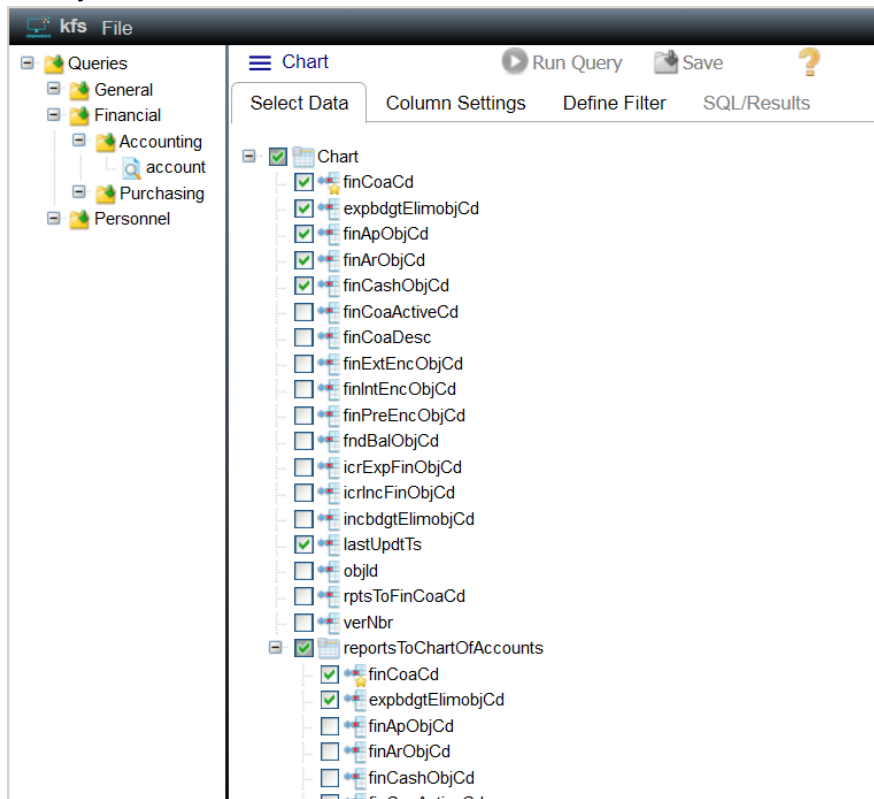
## Node.js Oracle ORM for Kualiti Financials



## Select Desired Model Columns

After the starting model is selected a model hierarchy will be displayed – choose the desired columns you want. Starred columns are primary key columns

## Node.js Oracle ORM for Kuali Financials



## Column Setting

Customize the selected column settings as desired 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 automatically generated. The **Custom** field allows the designer the flexibility to add any 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 specified order.

Column Index	Column Name	Label	Function	Sort Pos	Desc	Custom
1.	finCooCd	Chart		1	<input type="checkbox"/>	
2.	expbdgtElimobjCd				<input type="checkbox"/>	
3.	finApObjCd				<input type="checkbox"/>	
4.	finArObjCd				<input type="checkbox"/>	
5.	finCashObjCd				<input type="checkbox"/>	
6.	lastUpdtTs	Last Update		2	<input checked="" type="checkbox"/>	
7.	reportsToChartOfAccounts->finCooCd				<input type="checkbox"/>	
8.	reportsToChartOfAccounts->expbdgtElimobjCd				<input type="checkbox"/>	

## Define Filter

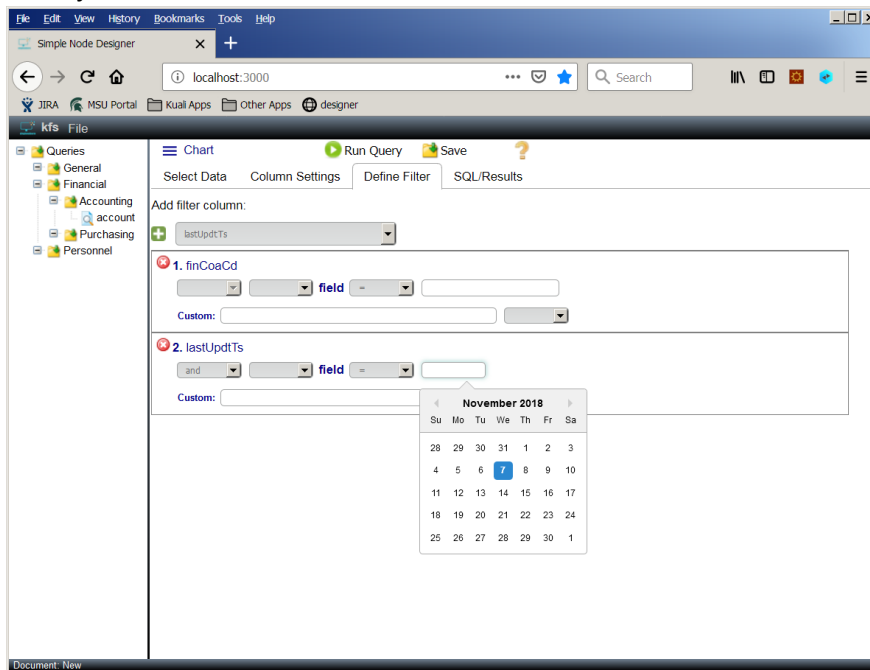
A where clause is required for all query documents. The Define Filter tab is where the designer builds the where clause. Where columns entries will be selected from the column selections, so you must ensure that desired filter columns are selected in the column selection tree. To add a filter entry,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

Add filter column: + finCooCd Add Button

- finCooCd
- expbdgtElimobjCd
- finApObjCd
- finArObjCd
- finCashObjCd
- lastUpdtTs
- reportsToChartOfAccounts.finCooCd
- reportsToChartOfAccounts.expbdgtElimobjCd



## Node.js Oracle ORM for Kuali Financials



Each filter line allows for selection of (where appropriate)

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 ))))

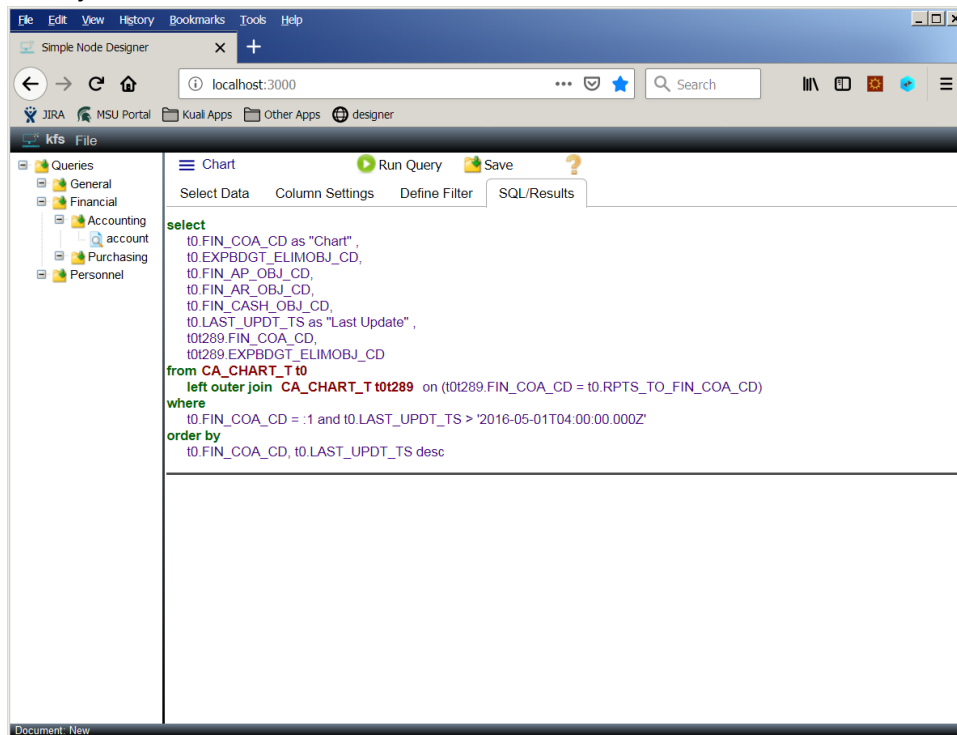
If a comparison value field is left empty is assumed that that field will be populated by a bind parameter when the document is run.

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

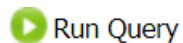
Once a filter has been defined the user can see the generated SQL in the SQL/Results tab:

## Node.js Oracle ORM for Kuali Financials



## 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 'Search Parameters' dialog box is shown. It has a title bar with the text 'Search Parameters'. Inside, there is a 'Result Format:' dropdown menu set to 'object graph'. Below it are two checkboxes: 'Distinct' and 'Validity Check Only', both of which are unchecked. There is a text input field labeled 'finCooCd:' with the value 'EA' entered. At the bottom of the dialog are two buttons: 'OK' and 'CANCEL'.

The parameter entry dialog also allows the user other options:

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

## Node.js Oracle ORM for Kuali Financials

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

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

### Object Graph Result:



The screenshot shows the SQL/Results tab of a query tool. The top bar includes a hamburger menu, 'Chart', 'Run Query', 'Save', and a help icon. Below the bar are tabs for 'Select Data', 'Column Settings', 'Define Filter', and 'SQL/Results'. The 'SQL/Results' tab is active, displaying a SQL query and its corresponding object graph result.

```
select
t0.FIN_COA_CD as "Chart",
t0.EXPBDGT_ELIMOBJ_CD,
t0.FIN_AP_OBJ_CD,
t0.FIN_AR_OBJ_CD.
```

```
{
  "_model_": "Chart",
  "FinCoeCd": "EA",
  "expbdgtElimobjCd": "1209",
  "finApObjCd": "9041",
  "finArObjCd": "8118",
  "finCashObjCd": "8000",
  "lastUpdtTs": "2009-07-01T05:00:00.000Z",
  "reportsToChartOfAccounts": {
    "_model_": "Chart",
    "FinCoeCd": "IU",
    "expbdgtElimobjCd": " "
  }
}
```

### Result Set Result:



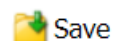
The screenshot shows the SQL/Results tab of a query tool. The top bar includes a hamburger menu, 'Chart', 'Run Query', 'Save', and a help icon. Below the bar are tabs for 'Select Data', 'Column Settings', 'Define Filter', and 'SQL/Results'. The 'SQL/Results' tab is active, displaying a SQL query and its corresponding result set result.

```
select
t0.FIN_COA_CD
```

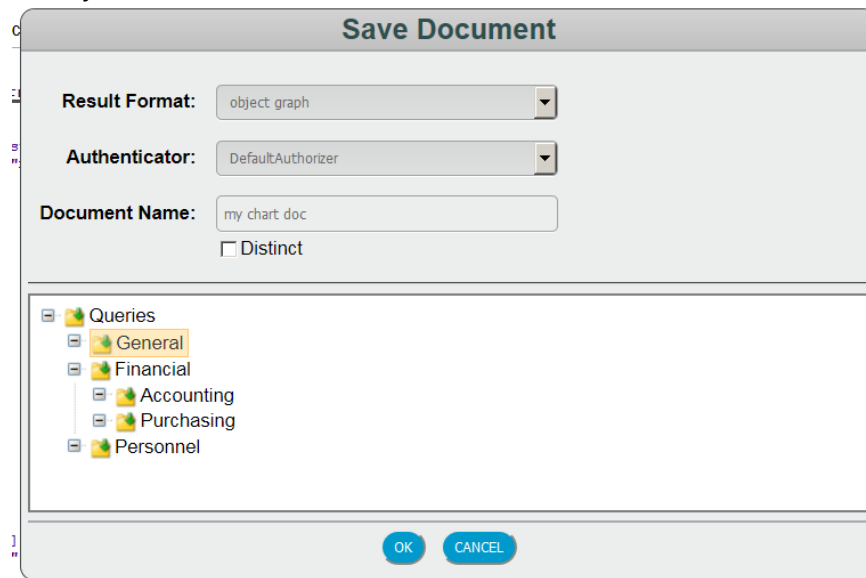
```
{
  "result": {
    "metaData": [
      {
        "name": "FIN_COA_CD"
      },
      {
        "name": "EXPBDGT_ELIMOBJ_CD"
      },
      {
        "name": "FIN_AP_OBJ_CD"
      },
      {
        "name": "FIN_AR_OBJ_CD"
      },
      {
        "name": "FIN_AP_OBJ_CD"
      },
      {
        "name": "FIN_AR_OBJ_CD"
      },
      {
        "name": "FIN_CASH_OBJ_CD"
      }
    ],
    "rows": [
      [
        "EA",
        "1209",
        "9041",
        "8118",
        "9041",
        "8118",
        "8000"
      ]
    ]
  }
}
```

## Saving the Query

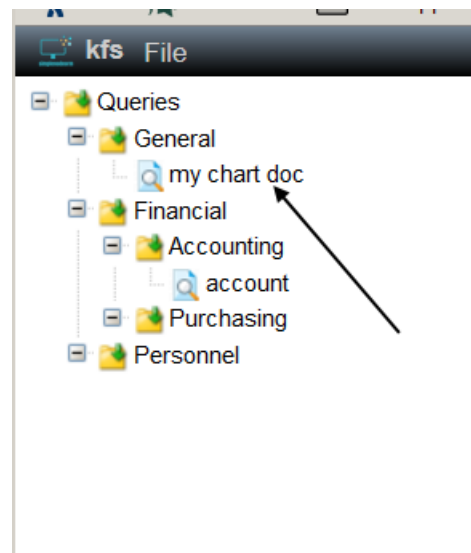
Click the save button to **Save** the Query Document



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:



## Editing and Deleting Existing Query Documents

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

## Node.js Oracle ORM for Kualiti Financials

