



Simple Node Query Designer

Rob Tucker
11/07/2018

Contents

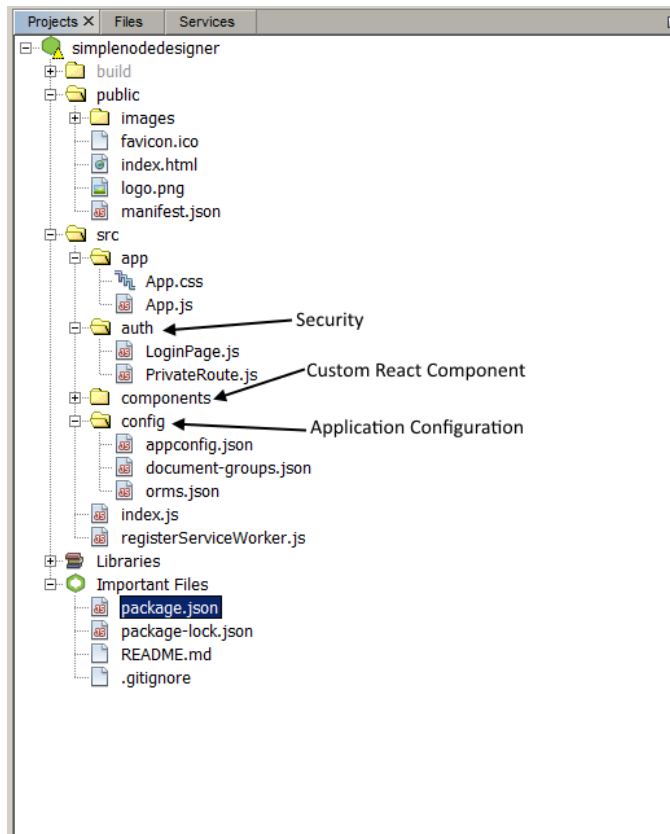
Introduction	2
Project Layout	2
Application Configuration	2
appconfig.json	2
document-groups.json	3
orms.json	4
Creating a Query Document	4
Login	4
Select Starting Model	4
Select Desired Model Columns	5
Column Setting	6
Define Filter	7
View Generated SQL	8
Running the Query	9
Saving the Query	10
Editing 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 documents 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",
    "modelselectdefault": "Select starting model...",
    "adddocument": "Add Document",
    "setupmenuname": "Setup",
    "filemenuname": "File",
    "newmenuname": "New",
```

Node.js Oracle ORM for Kuali Financials

```
"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 access will be granted to everyone.

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. On the ORM server the root path for document storage is set in the ORM configuration. Documents 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
        },
        {
          "title": "Purchasing",
          "key" : "grp4",
          "isLeaf": false
        }
      ]
    }
  ]
}
```

Node.js Oracle ORM for Kuali Financials

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

orms.json

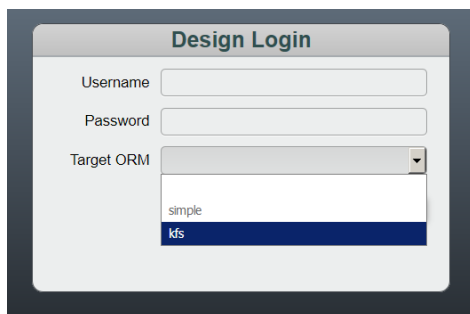
This Query Designer supports multiple ORM installations. The available ORMs are define as shown below. defaultUsername and defaultPassword are optional, if they exists they will auto-populate the associated login dialog fields.

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

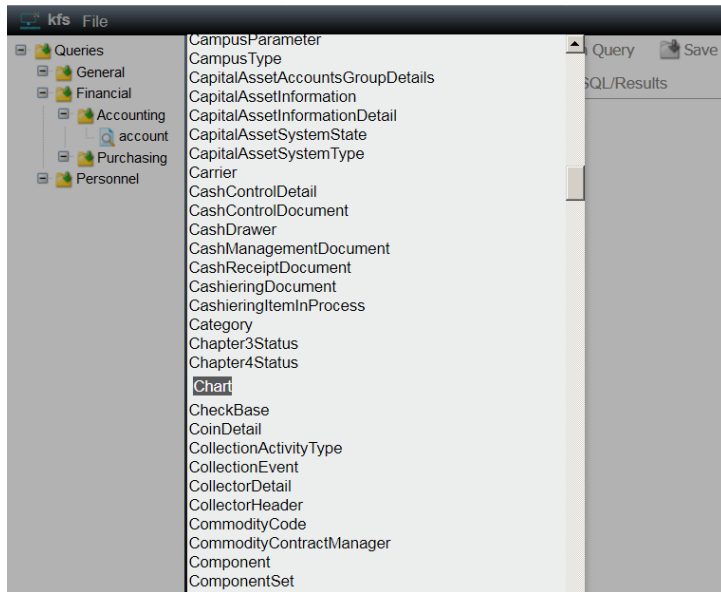
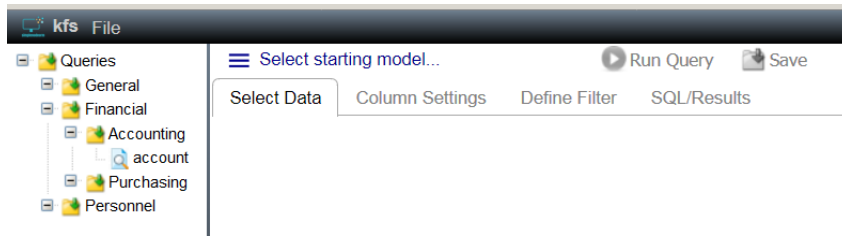


Select Starting Model

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

 [Select starting model...](#)

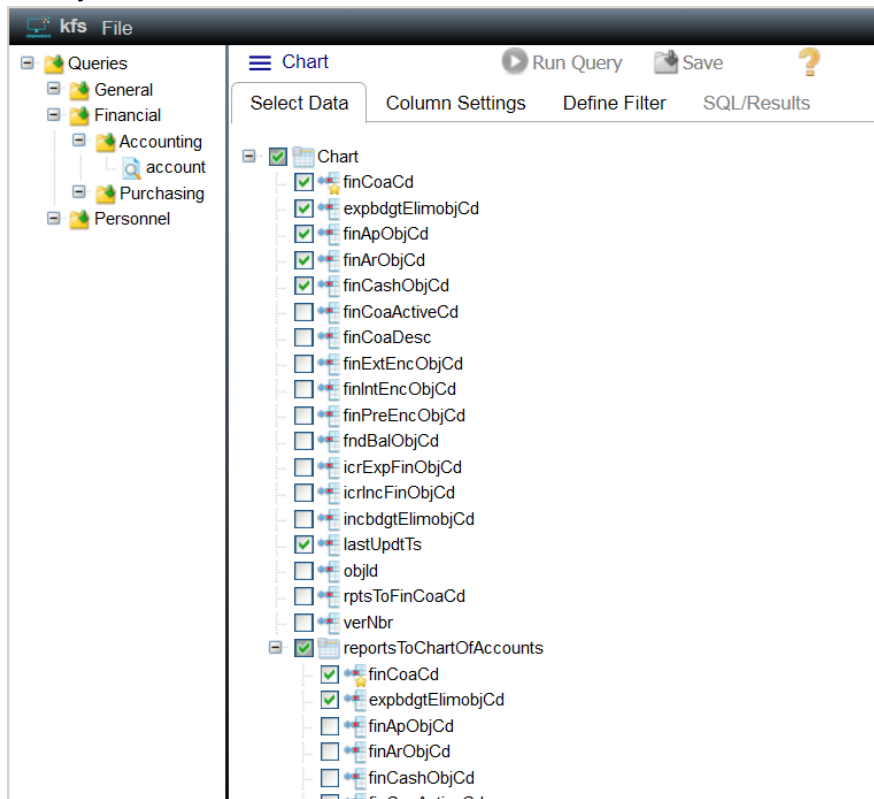
Node.js Oracle ORM for Kualu Financials



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 order specified.

Column	Label	Function	Sort Pos	Desc	Custom
1. finCocCd	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->finCocCd				<input type="checkbox"/>	
8. reportsToChartOfAccounts->expbdgtElimobjCd				<input type="checkbox"/>	

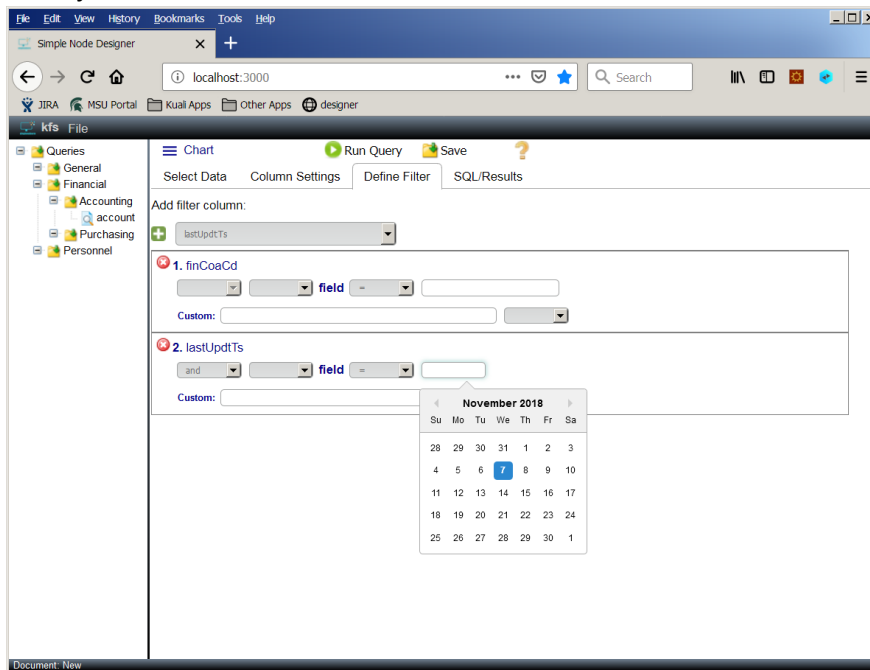
Define Filter

A where clause is required for all query documents. The **Define Filter** tab is where the designer builds the where clause. Where column 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: finCocCd Add Button

- finCocCd
- expbdgtElimobjCd
- finApObjCd
- finArObjCd
- finCashObjCd
- lastUpdtTs
- reportsToChartOfAccounts.finCocCd
- 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 - (, (, ((, (((, (((
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 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:

The screenshot shows the Simple Node Designer application. The top menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The main toolbar contains navigation icons (back, forward, refresh, home), a search bar, and a status bar. The left sidebar shows a tree view of data sources: Queries, General, Financial, Accounting, account, Purchasing, and Personnel. The main workspace is divided into two tabs: Chart and SQL/Results. The Chart tab is active, displaying a SQL query:

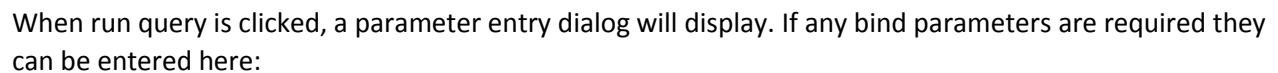
```

select
  t0.FIN_COA_CD as "Chart",
  t0.EXPBDGT_ELIMOBJ_CD,
  t0.FIN_AP_OBJ_CD,
  t0.FIN_AR_OBJ_CD,
  t0.FIN_CASH_OBJ_CD,
  t0.LAST_UPDT_TS as "Last Update",
  t0t289.FIN_COA_CD,
  t0t289.EXPBDGT_ELIMOBJ_CD
from CA_CHART_T t0
left outer join CA_CHART_T t0t289 on (t0t289.FIN_COA_CD = t0.RPTS_TO_FIN_COA_CD)
where
  t0.FIN_COA_CD = :1 and t0.LAST_UPDT_TS > '2016-05-01T04:00:00.000Z'
order by
  t0.FIN_COA_CD, t0.LAST_UPDT_TS desc

```

The bottom status bar indicates "Document: New".

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



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

- 9

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' (with a play icon), 'Save' (with a floppy disk icon), and a help icon. Below the bar are tabs: 'Select Data', 'Column Settings', 'Define Filter', and 'SQL/Results' (which is active). The main area displays a SQL query: `select t0.FIN_COA_CD as "Chart", t0.EXPBDGT_ELIMOBJ_CD, t0.FIN_AP_OBJ_CD, t0.FIN_AR_OBJ_CD;`. Below the query is a JSON object graph representing the result set. The object has a `_model` property set to `"Chart"`, a `FinCoeCd` property set to `"EA"`, an `expbdgtElimobjCd` property set to `"1209"`, a `finApObjCd` property set to `"9041"`, a `finArObjCd` property set to `"8118"`, a `finCashObjCd` property set to `"8000"`, a `lastUpdtTs` property set to `"2009-07-01T05:00:00.000Z"`, and a `reportsToChartOfAccounts` property set to an object with `_model` `"Chart"`, `FinCoeCd` `"IU"`, and `expbdgtElimobjCd` `" "`.

```
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:



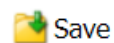
The screenshot shows the SQL/Results tab of a query tool, similar to the previous one. The SQL query is `select t0.FIN_COA_CD;`. The result set is displayed as a JSON object with a `result` property containing a `metaData` array and a `rows` array. The `metaData` array lists the column names: `FIN_COA_CD`, `EXPBDGT_ELIMOBJ_CD`, `FIN_AP_OBJ_CD`, `FIN_AR_OBJ_CD`, `FIN_AP_OBJ_CD`, `FIN_AR_OBJ_CD`, `FIN_AR_OBJ_CD`, and `FIN_CASH_OBJ_CD`. The `rows` array contains a single row of values: `"EA"`, `"1209"`, `"9041"`, `"8118"`, `"9041"`, `"8118"`, `"8118"`, and `"8000"`.

```
select
t0.FIN_COA_CD

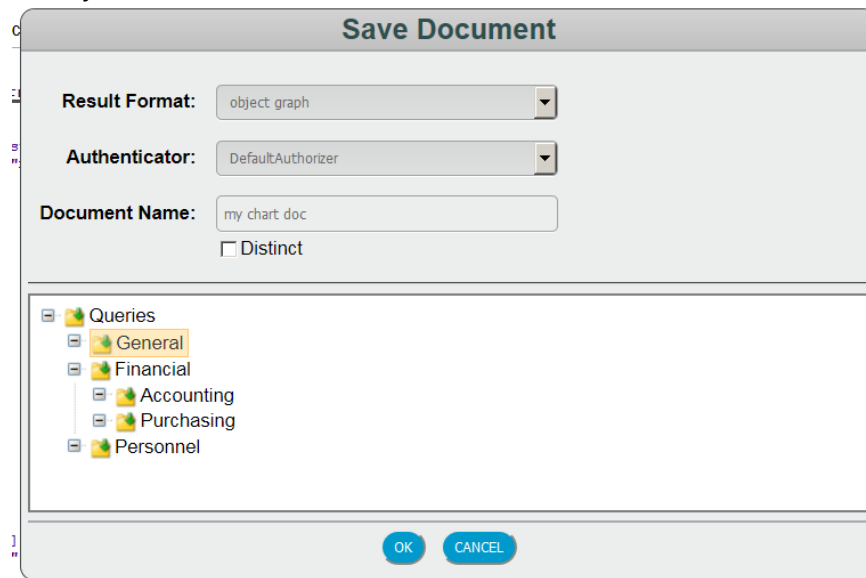
{
  "result": {
    "metaData": [
      {
        "name": "FIN_COA_CD"
      },
      {
        "name": "EXPBDGT_ELIMOBJ_CD"
      },
      {
        "name": "FIN_AP_OBJ_CD"
      },
      {
        "name": "FIN_AR_OBJ_CD"
      },
      {
        "name": "FIN_AP_OBJ_CD"
      },
      {
        "name":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

Saving the Query

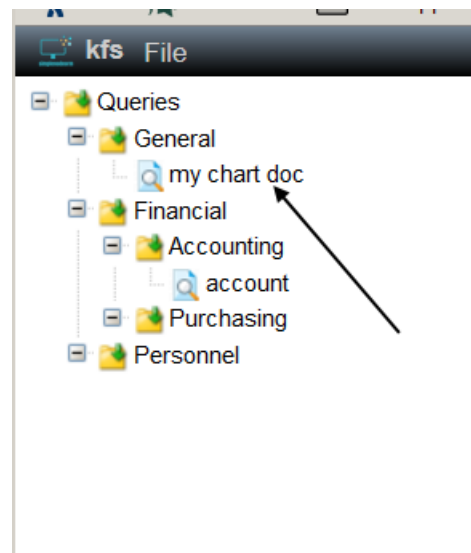
Click the save button to **Save** the Query Document



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

