Configuring PoS dashboards

Predefined dashboard are in package PAV4\_GENERIC\_POS\_DASHBOARD. This package can be overridden as any Cassiopae package.

Source code for dashboards are available here:

<https://front.cassiopae.com/POSINNO/sql/PAV4_GENERIC_POS_DASHBOARD.SQL>

<https://front.cassiopae.com/POSINNO/sql/PAV4_GENERIC_POS_CHART.SQL>

# Get the list of dashboards for current user

Upon login the server will call this SQL function:

|  |
| --- |
| FUNCTION GETDASHBOARDS( suticode utilisateur.uticode%type, p\_params varchar2, p\_extparams varchar2) RETURN CLOB  IS  lsql CLOB:='';  crlf CHAR(2) := '  ';  BEGIN  lsql := lsql || '{' || crlf;  lsql := lsql || ' "user": "' || suticode || '",' || crlf;  lsql := lsql || ' "dashboards": [' || crlf;  lsql := lsql || ' { "title": "Dashboard 1", "id": "MAIN" },' || crlf;  lsql := lsql || ' { "title": "Dashboard 2", "id": "1" }' || crlf;  lsql := lsql || ' ]  }  ';  RETURN lsql;  END GETDASHBOARDS; |

This function returns 2 dashboards, “Dashboard 1” and “Dashboard 2” (it can return as many as you require).

"/dashboard/MAIN” will be called whenever “Dashboard 1” is requested, and also on the main page since it is the first in the list.

When called the server will execute function PAV4\_GENERIC\_POS\_DASHBOARD.F\_MAIN.

# Get the dashboard layout

Function F\_MAIN returns the main dashboard layout. Here:

* 4 charts on first line
* 3 on second line
* 2 on third line

|  |
| --- |
| FUNCTION F\_MAIN(  suticode utilisateur.uticode%type,  P\_PARAMS PAV4\_GENERIC\_POS.T\_PARAMS)  RETURN CLOB  IS  lsql CLOB:='{ "layout": [  [  {"url":"/chart/KPIDEALINPROGRESS", "layoutClass": "col-lg-3 col-xs-6"},  {"url":"/chart/KPICOMMISSION", "layoutClass": "col-lg-3 col-xs-6"},  {"url":"/chart/KPIPROPOSALATTENTION", "layoutClass": "col-lg-3 col-xs-6"},  {"url":"/chart/KPIRENEGOCIATION", "layoutClass": "col-lg-3 col-xs-6"}  ],  [  {"url":"/chart/DEALINPROGRESS"},  {"url":"/chart/COMMISSIONPERMONTH"},  {"url":"/chart/SALESPERMONTH2"}  ],  [  {"url":"/chart/HTMLARKLE", "layoutClass": "col-lg-8 col-xs-8"},  {"url":"/chart/CONTACTLIST", "layoutClass": "col-lg-4 col-xs-4"}  ]  ]}';  BEGIN  RETURN lsql;  END F\_MAIN; |

Layout class are the Bootstrap CSS that pilot the responsiveness of the dashboard.

* “col-lg-3” means “In large layout (PC), use 3 out of 12 columns for that chart e.g. ¼ of the screen”
* “col-xs-6” means “In x-small layout (phone), use 6 out of 12 e.g. ½ of the screen “
* “col-md-8” means “In medium layout (tablet), use 8 out of 12 e.g. 2/3 of the screen “

Each chart is defined by its REST url. For example /chart/KPIDEALINPROGRESS will call SQL Function F\_KPIDEALINPROGRESS in package PAV4\_GENERIC\_POS\_CHART.

# Get CHART

|  |
| --- |
| FUNCTION F\_KPIDEALINPROGRESS(  suticode utilisateur.uticode%type,  P\_PARAMS PAV4\_GENERIC\_POS.T\_PARAMS)  RETURN CLOB  IS  lsql CLOB:='';  res number := 0.0;  BEGIN  select count ( distinct (DIN.DOSID)) into res from V\_DEAL DPR, DPRINTERVENANT DIN  WHERE DPR.DPRVERSION IN ('NEGO','FIN')  AND DPR.DOSID = DIN.DOSID  AND DPR.DPRVERSION = DIN.DPRVERSION  AND DIN.UTICODE = NVL(suticode, 'DEALER1');  --select 23.0 into res from dual;  lsql := lsql || '{  "type": "KPI",  "parameters": {  "type": "green",  "title": "Deals in progress",  "icon": "fa-thumbs-o-up",  "link": "#mydeals2",  ';  lsql := lsql || '"number": ' || res;  lsql := lsql || '}  }';  RETURN lsql;  END |

This SQL Function returns very simple JSON that describes the chart to draw.

|  |
| --- |
| {  "type": "KPI",  "parameters": {  "type": "green",  "title": "Deals in progress",  "icon": "fa-thumbs-o-up",  "link": "#mydeals2",  "number": 0}  } |

All other charts are working on a similar pattern. See examples in the SQL files.

Existing charts (see code for examples)

* KPI & KPI2
* Doughnut
* Bar
* Line
* Area
* HTML
* ContactList
* IMAGE

Possible charts

* Radar?
* Bubble?
* Maps?