

Pentaho Ctools : tips & tricks

trucs et astuces pour le développement de vos dashboards
Pentaho

Présenté par Benoît Mouchard



Liste déroulante et « Tous »

- Besoin : filtrer ou non selon un élément id/libellé parmi une liste provenant d'une table de référence

The screenshot displays a web application interface. On the left, a dropdown menu is open, showing a list of fruit names: (Tous), (Tous), banane, ananas, pomme, abricot, cerise, fraise, poire, and pêche. The first two items are highlighted in blue. In the center, a table with two columns is visible. The first column is labeled 'Pâtisserie' and the second 'Parfum'. The table contains the following data:

Pâtisserie	Parfum
tarte	abricot
tarte	ananas
charlotte	ananas
sablé	cerise

On the right, a 'Properties / Advanced Properties' panel is shown. It has a table with two columns: 'Property' and 'Value'. The 'Datasource' property is highlighted in blue and has the value 'fruitQuery'.

Property	Value
Name	selectFruit
Parameter	-
Listeners	[]
Parameters	[]
jQuery Plugin	-
Value as id	False
Datasource	fruitQuery
Values array	[]
HtmlObject	colFruit

Liste déroulante et « Tous »

Requête du composant « select »

```
SELECT id_fruit,  
       label  
FROM fruit
```



```
SELECT -1 AS id_fruit,  
       '(Tous)' AS label  
FROM fruit  
UNION  
SELECT id_fruit,  
       label  
FROM fruit
```

Requête du tableau

```
SELECT patisserie.label AS "Pâtisserie",  
       fruit.label AS "Parfum"  
FROM patisserie  
JOIN fruit using(id_fruit)  
WHERE patisserie.id_fruit = ${paramFruit}
```



```
SELECT patisserie.label AS "Pâtisserie",  
       fruit.label AS "Parfum"  
FROM patisserie  
JOIN fruit using(id_fruit)  
WHERE patisserie.id_fruit = ${paramFruit}  
OR ${paramFruit} = -1 ;
```

Export Excel d'une query existante

- Ouvrir le fichier .cda  correspondant au dashboard
- Sélectionner la query et remplir les paramètres , puis Query URL

• Filename: /public/test/test.cda [About](#)

tarteQuery 1

Exporter au format .xls Query URL Cacher 

paramFruit:

-1

Show 10 elements Search:

Pâtisserie	Parfum
charlotte	ananas
sablé	cerise
tarte	abricot
tarte	ananas

Export Excel d'une query existante

On obtient l'url correspondant à la requête



http://**mon_serveur**/pentaho/plugin/cda/api/doQuery
?path=/**chemin_solution**/**mon_cda.cda**
&data**AccessId**=**tarteQuery**
¶m**paramFruit**=-1

Export Excel d'une query existante

- Ajouter les paramètres suivants

`&outputType=xls`

`&settingattachmentName=tartes.xls`

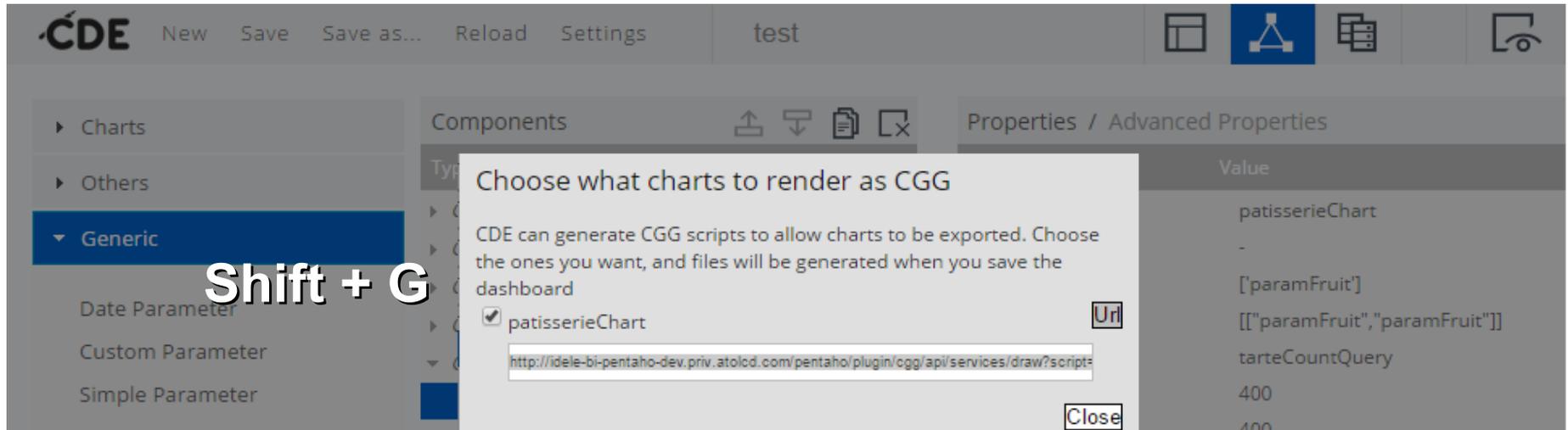
- Url directement appelable depuis un lien

Type Structure	Année	Région	Dépt.	Ressource	Module	Nb utilisateurs	Nb connexions	
SGA	2016	PARIS	ILE-DE-FRANCE	ACCUEIL	Accueil	1	96	
SGA	2016	PARIS	ILE-DE-FRANCE	AN_COMPARAISSON_EVOLUTION_NATIONALE	Agrégats nationaux	1	8	

- Url appelable depuis l'extérieur avec une authentification par url

Export des graphiques en images

- Les graphiques affichés dans les dashboards : format svg (dessin vectoriel), un format standard mais nécessitant une conversion en image pour être exploitable facilement
- Grâce au CGG (Community Graphics Generator), on peut générer un script Javascript permettant la conversion



The screenshot shows the CDE (Community Desktop Environment) interface. The top menu bar includes 'CDE', 'New', 'Save', 'Save as...', 'Reload', 'Settings', and 'test'. The left sidebar shows a tree view with 'Charts' and 'Others' expanded, and 'Generic' selected. A dialog box titled 'Choose what charts to render as CGG' is open, displaying a list of charts with checkboxes. The 'patisserieChart' is checked. Below the list is a text input field containing a URL: 'http://idele-bi-pentaho-dev.priv.atolcd.com/pentaho/plugin/cgg/api/services/draw?script='. The dialog also has 'Url' and 'Close' buttons.

Shift + G

Value
patisserieChart
-
[paramFruit]
[["paramFruit","paramFruit"]]
tarteCountQuery
400
400

Export des graphiques en images

- Url générée

[http://mon_serveur/pentaho/plugin/cgg/api/services/draw?script=
/chemin_solution/mon_gaph.js&outputType=png](http://mon_serveur/pentaho/plugin/cgg/api/services/draw?script=/chemin_solution/mon_gaph.js&outputType=png)

- Ajouter les paramètres nécessaires au graphique

[¶mparamFruit=-1](#)

Export des graphiques en images

- **Avantage:**
 - Facilité de mise en œuvre
 - Url directement appelable depuis un lien
 - Accès externe possible en authentification par url
- **Limitation**
 - le script n'utilise pas le graphique affiché dans le dashboard en svg, mais fait un nouveau rendu
 - **les customisations codées ne sont pas reportées!**



Export des graphiques en images

- Alternative : convertir le svg affiché à l'écran en image
 - **Avantage:**
 - Sauvegarde exacte du graphique affiché
 - **Inconvénient**
 - Développement spécifique nécessaire, avec utilisation de différentes librairies Javascript
 - Faire un rendu du svg dans un canvas Html (librairie canvg github.com/gabelerner/canvg)
 - Convertir le canvas en Blob (librairie canvas-toBlob.js github.com/eboyjr)
 - Sauvegarder le Blob en fichier (FileSaver.js même auteur)

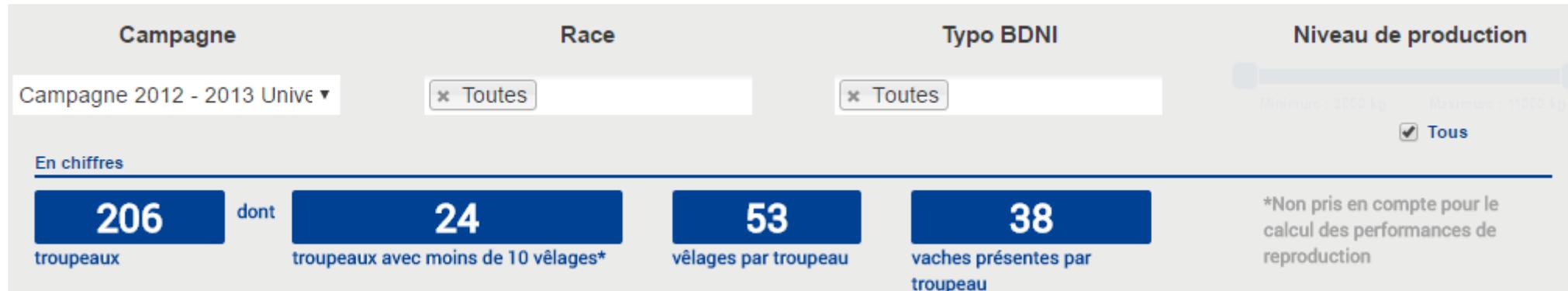
Exemple

<https://drive.google.com/file/d/0BwocIDK0A2zXVXNtUllvcHNkTnM/view?usp=sharing>



Filtres réutilisables

- Couramment nous avons plusieurs dashboards ayant des filtres en commun. Les développer une seule fois et les réutiliser est un gain de temps et gage de maintenabilité

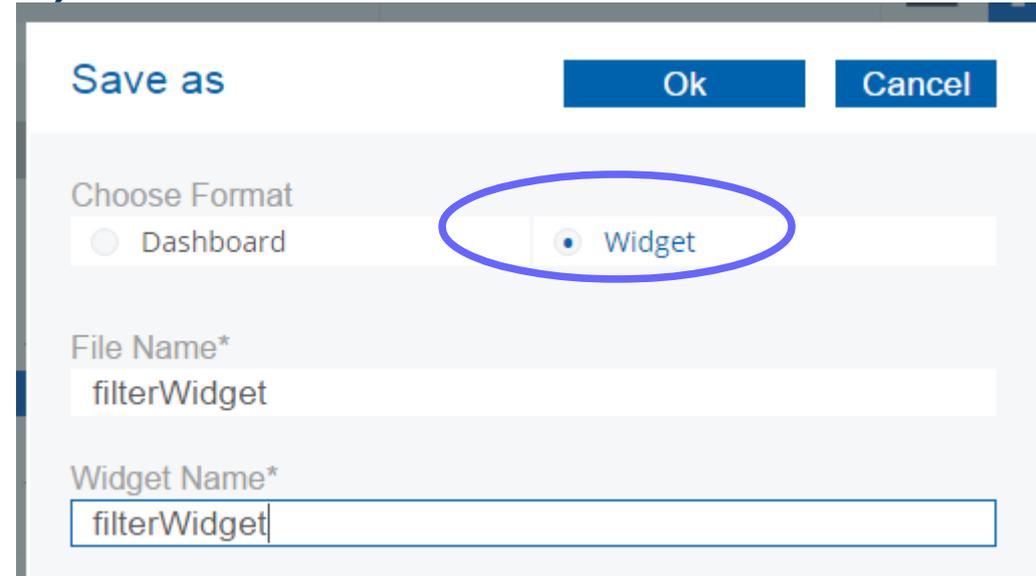


- La solution Ctools : la création et utilisation de widget

Filtres réutilisables (widget)

Le Principe

- Créer un dashboard avec le contenu nécessaire au filtre et le sauvegarder en widget



The screenshot shows a 'Save as' dialog box with the following fields and options:

- Save as** (title)
- Ok** and **Cancel** buttons
- Choose Format** section with two radio buttons: Dashboard and Widget (the 'Widget' option is circled in blue).
- File Name*** text box containing 'filterWidget'.
- Widget Name*** text box containing 'filterWidget'.

- Syntaxe particulière au widget, toute référence à un paramètre, un composant ou un élément HTML du widget doit-être de la forme :
`#{p:parameterName}` , `#{c:componentName}`, `#{h:htmlObject}`

Filtres réutilisables (widget)

- Définir les paramètres accessible du widget

Settings

Ok **Cancel**

Title
filterWidget

Widget Name
filterWidget

Author
Insert Text...

Dashboard Type
bootstrap

RequireJS Support

Widget Parameters

paramOutWidget

paramInWidget

Components		Properties / Advanced Properties	
Type	Name	Property	Value
▼ Group	Generic	Name	paramOu
Simple Parameter	paramOutWidget	Property value	yoOut
Simple Parameter	paramInWidget	Bookmarkable	False
▶ Group	Others	Public	True

Filtres réutilisables (widget)

- Utiliser le widget dans le dashboard

The screenshot shows the CDE (Customer Data Engine) interface. The top bar includes the CDE logo, menu items (New, Save, Save as..., Reload, Settings), and the current workspace name 'test *'. On the left, a sidebar lists various widget categories: Charts, Others, Generic, Scripts, Selects, CDF Core Functionality, and Widgets (which is expanded to show 'filterWidget' and 'widgetSampleWidget'). The main area is divided into three panels: 'Components', 'Properties / Advanced Properties', and a central workspace. The 'Components' panel is a table listing widget types and their names. The 'Properties / Advanced Properties' panel is a table showing the configuration for the selected 'filterWidget'.

Type	Name
▶ Group	Selects
▶ Group	Generic
▶ Group	Others
▼ Group	Widgets
BmoTestWidget	widgetComponent
filterWidget	filterWidget
▶ Group	Charts

Property	Value
Name	filterWidget
Parameter paramInWidget	param1
Parameter paramOutWidget	param2
HtmlObject	filterContainer

Note : les widgets sont stockés dans /public/cde/widgets

Filtres réutilisables (widget)

- Communication dashboard ↔ widget

