

sage FRP 1000

Release Guide

Sage FRP 1000
Version 9.00 Release 2

Conception de tableau de bord

Juin 2021



Sommaire

PRESENTATION	4
Prérequis	4
Accès au concepteur de tableau de bord	6
Création d'un tableau de bord	7
LES VIGNETTES	10
Création d'une vignette	12
Les indicateurs	13
Les indicateurs de tendances	17
Les grilles	21
Les listes de valeurs	25
Les listes simples	27
Les graphiques « Camembert » ou « Donut »	29
Les graphiques « Histogramme »	32
LES FILTRES	35
Filtre global	35
Filtre entre vignettes	38

Présentation

Depuis les versions 9.00 Release 1 & 2, de nouvelles possibilités sont offertes dans les tableaux de bord :

- les vignettes indicateurs
- des liens hypertexte et passage de paramètres vers des écrans
- des grilles avec « flip » (bascule vers une grille de détail)
- ...

Ce guide permet d'appréhender ces nouveautés en se basant sur les exemples.

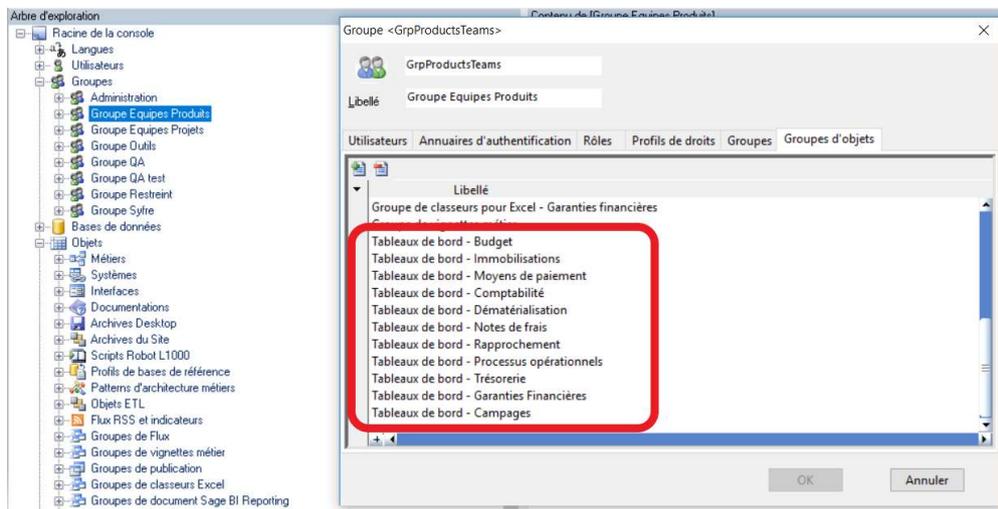
Prérequis

Il existe des prérequis à la création et à l'utilisation des tableaux de bord.

Les tableaux de bord sont accessibles uniquement en client léger, il convient donc d'installer un serveur web ainsi qu'un service correspondant au dossier et à l'application qui sera utilisée pour accéder au tableau de bord.

Il convient de créer un groupe d'utilisateurs dans lequel tous les groupes objets « Tableaux de bord » sont associés.

L'utilisateur qui se connecte au dossier doit être rattaché à ce groupe.



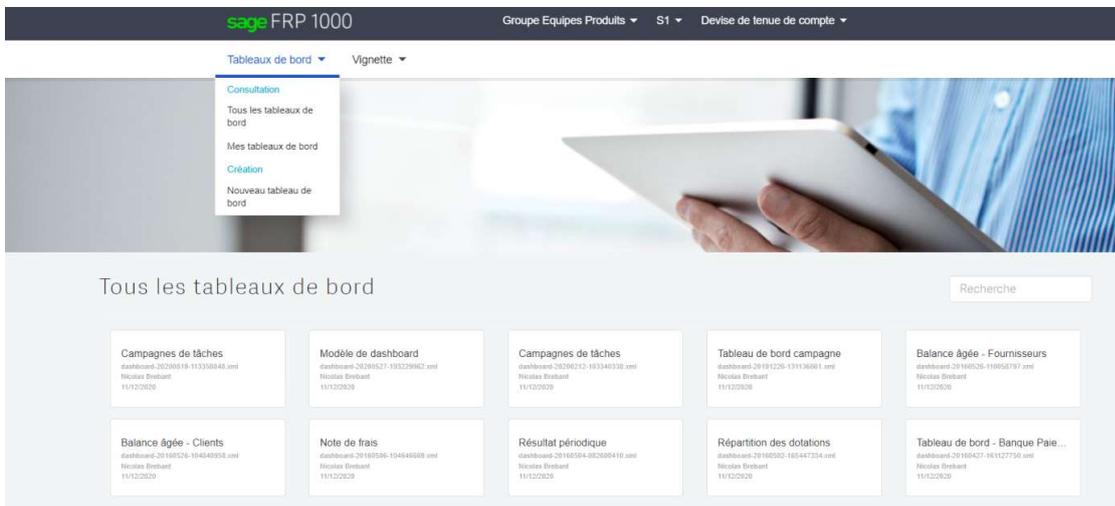
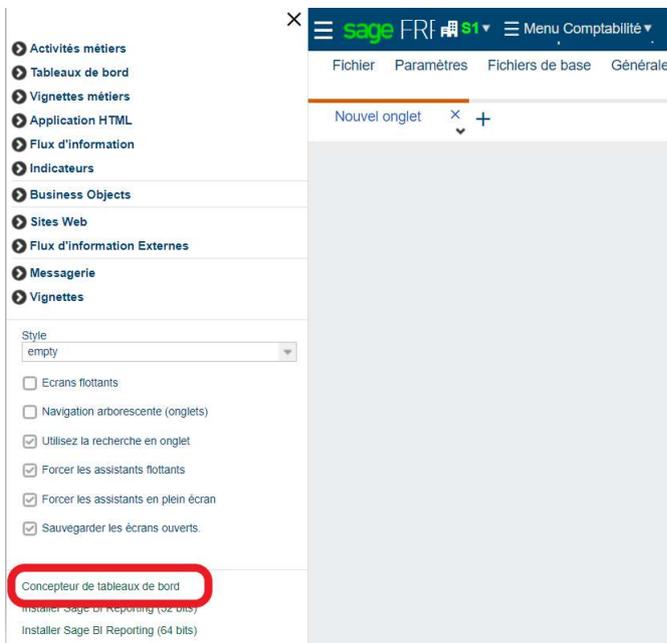
Lors du démarrage de l'application avec l'utilisateur, il faut sélectionner également le groupe ayant accès aux tableaux de bord.



Accès au concepteur de tableau de bord

Le concepteur de tableaux de bord est un outil permettant de concevoir des vignettes et des tableaux de bord présentant des données issues de vues métier ou de méthodes de services.

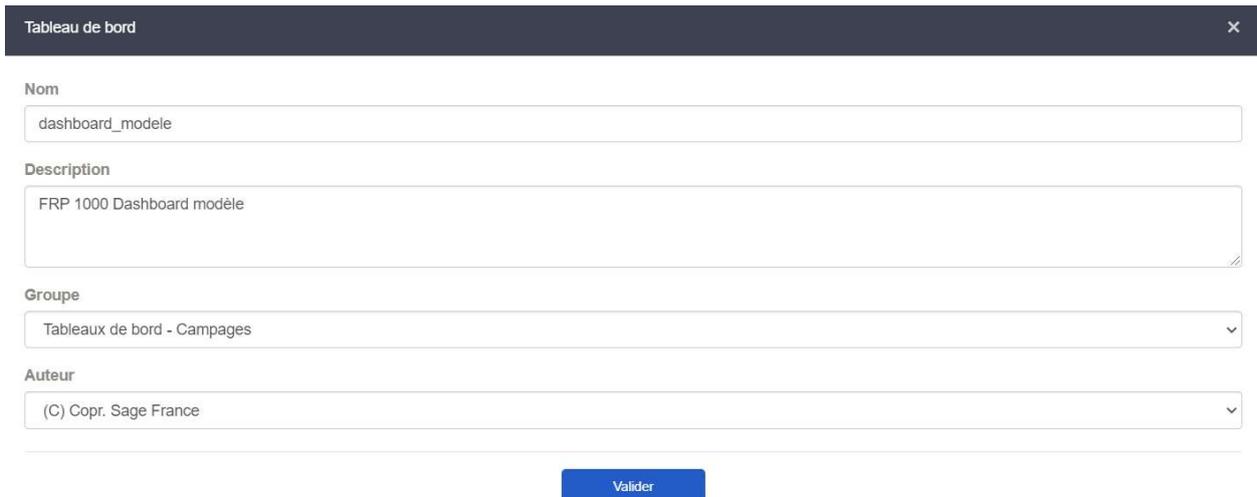
Le concepteur de tableaux de bord est accessible uniquement en web



Création d'un tableau de bord

La documentation sur les tableaux de bord se base sur le dashboard modèle livré en version 9.00 Release 2 (dashboard_modele - dashboard-20210224-173715323).

Ce tableau de bord contient tous les types de vignettes et le code correspondant pour chacune.



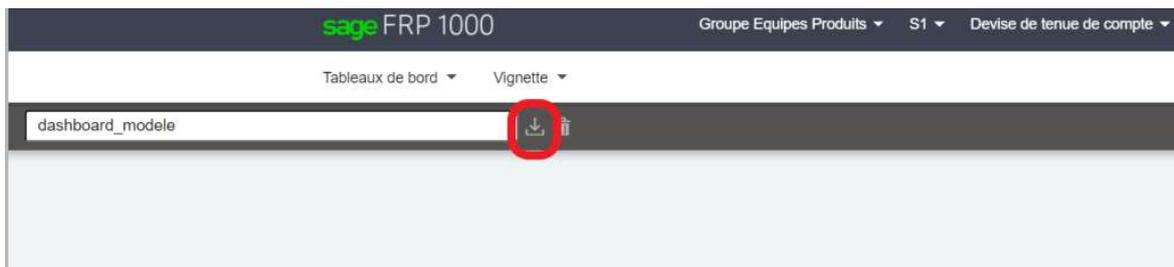
The screenshot shows a form titled "Tableau de bord" with a close button (X) in the top right corner. The form contains the following fields:

- Nom:** A text input field containing "dashboard_modele".
- Description:** A text area containing "FRP 1000 Dashboard modèle".
- Groupe:** A dropdown menu with "Tableaux de bord - Campagnes" selected.
- Auteur:** A dropdown menu with "(C) Copr. Sage France" selected.

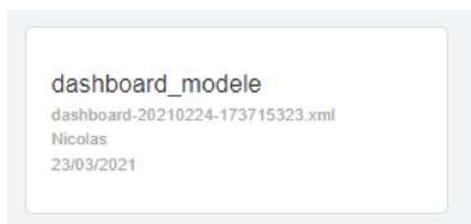
At the bottom center of the form is a blue button labeled "Valider".

Lors de la création d'un tableau de bord, il convient de renseigner un nom, une description, de sélectionner le groupe de tableau de bord concerné et de valider la création.

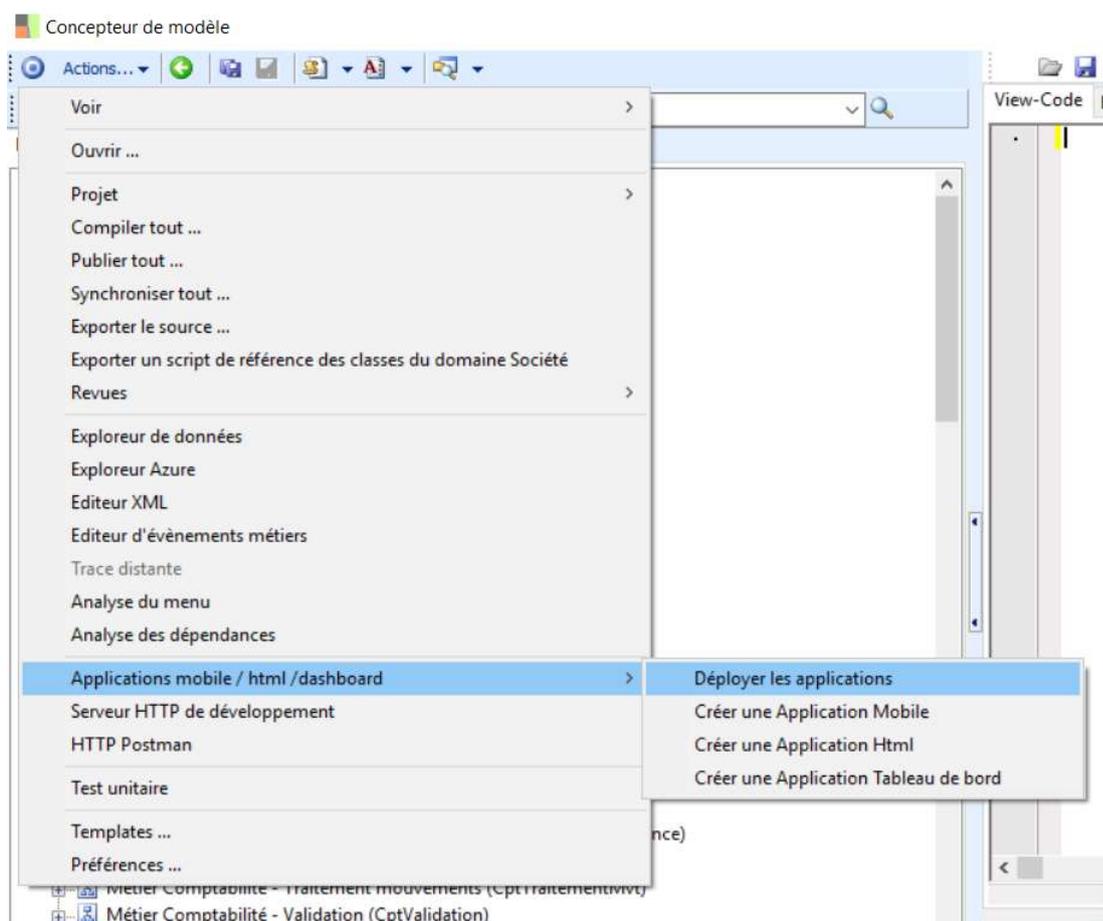
Le tableau de bord est créé, il est nécessaire de le publier (sauvegarder) puis de définir des vignettes qui seront utilisées dans le tableau de bord.



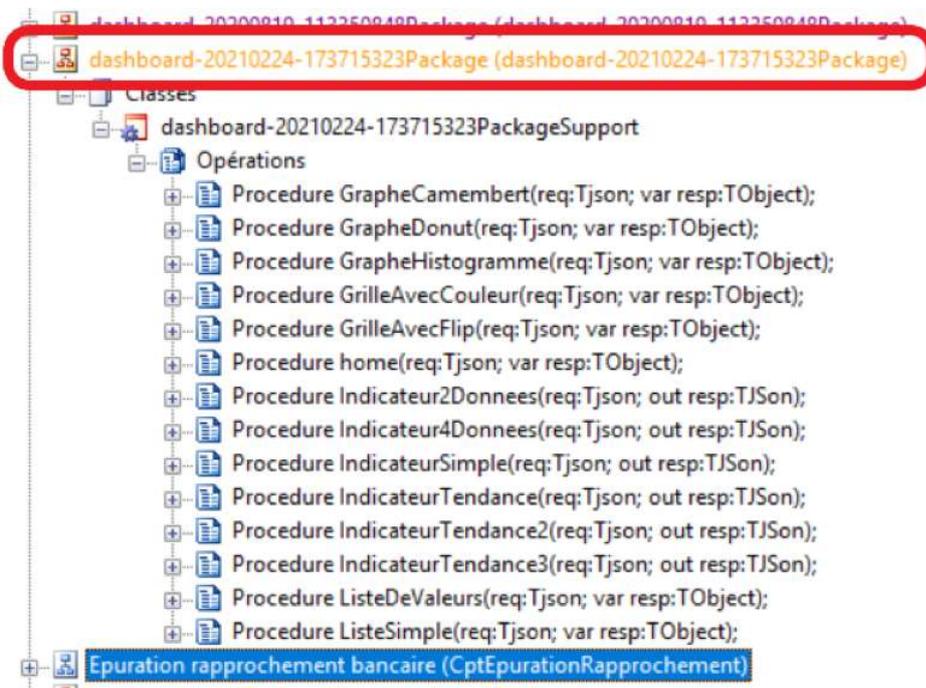
Le nom du tableau de bord est calculé par l'application, il est visible au niveau concepteur de tableau de bord.



Pour accéder aux opérations du tableau de bord, il convient de démarrer l'application L1000, de se rendre au niveau du concepteur de modèle et de lancer le déploiement des applications mobile / html / dashboard.



On accède alors aux différents tableaux de bord... et au paquet correspondant au tableau de bord « dashboard_modele » (dashboard-20210224-173715323).



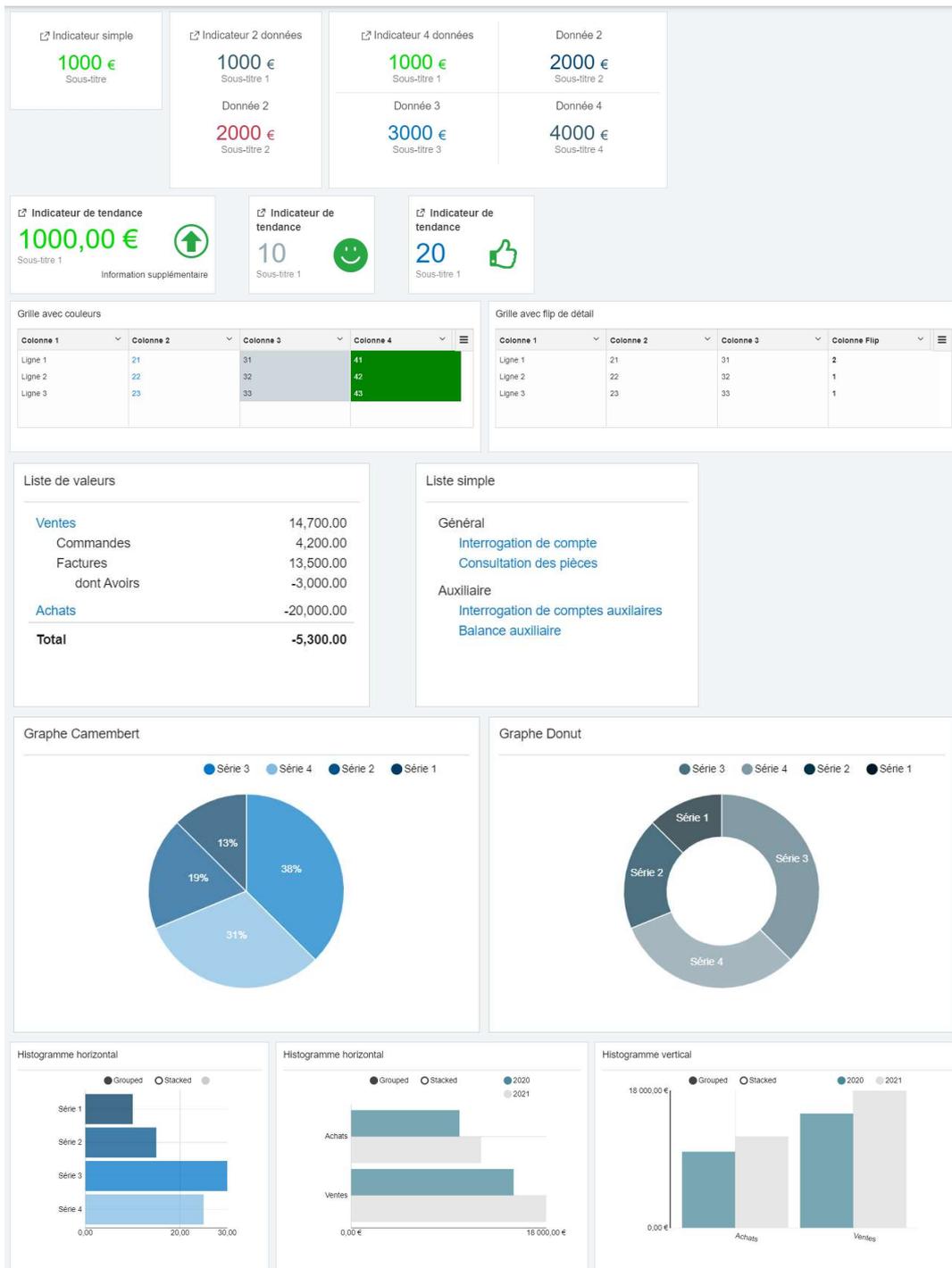
Dès que l'on procède à des ajouts ou à modifications des opérations, il convient de :

- Sauvegarder les modifications et le tableau de bord
- Publier le tableau de bord dans le référentiel
- Redémarrer le service correspondant

Les modifications sont alors disponibles au niveau de l'application en web et dans le concepteur de tableaux de bord.

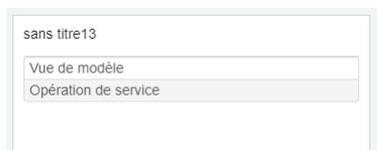
Les vignettes

Le tableau de bord se compose de vignettes, pour chaque vignette il est nécessaire de créer une opération ou une opération de service au niveau de la classe du tableau de bord.



Pour toutes les vignettes qui seront illustrées ci-après, les exemples présentent des résultats « fixes », il est nécessaire d'ajouter le code (variables, expressions, ...) qui alimentera les différents éléments.

L'ajout d'une vignette dynamique se fait par le bouton



Une vignette s'ajoute.

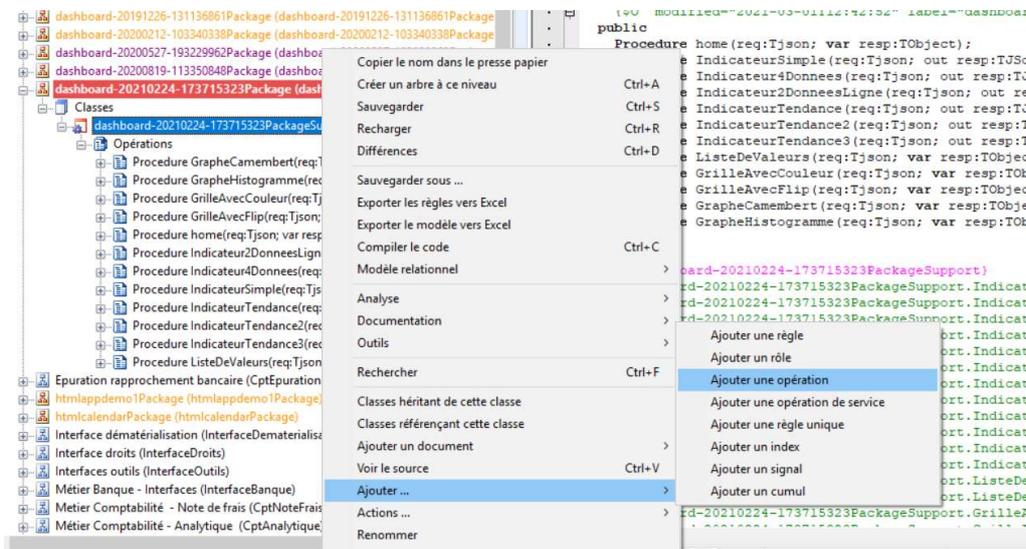
Il est possible de paramétrer une vue de modèle ou pour le cas qui nous concerne dans ce document une opération de service. Il faut sélectionner le type de vignette que l'on souhaite afficher, seules les opérations existantes correspondantes seront affichées. Il convient donc de créer préalablement les procédures au niveau du concepteur de modèle.

Les types d'opération (qualificatifs JSON) sont :

- Chart : les vignettes graphiques
- Tabulaire : les grilles
- KPI : les indicateurs
- Liste : les listes de valeurs

Création d'une vignette

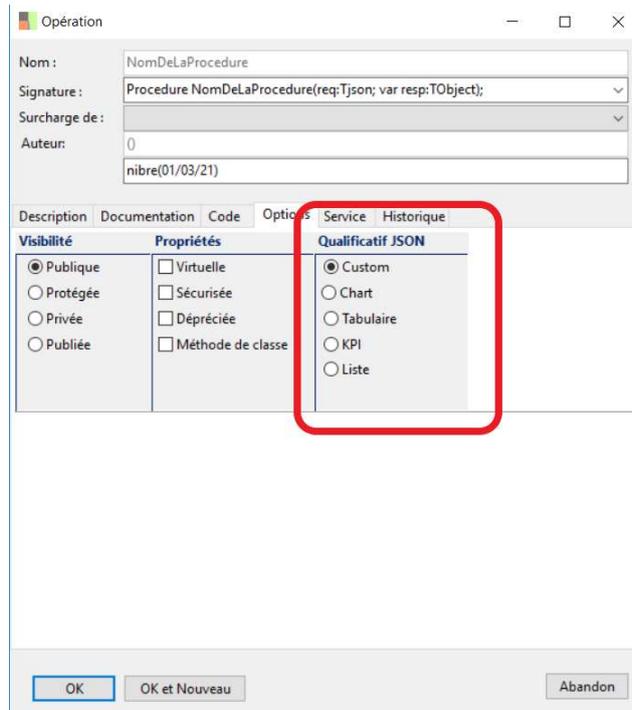
Il convient de créer une opération au niveau de la classe du tableau de bord :



Au niveau de la signature, il convient de renseigner un nom pour la procédure ainsi que les paramètres d'entrée :

Procédure NomDeLaProcédure(req:Tjson; var resp:TObject);

et de sélectionner le qualificatif JSON en fonction de la vignette à réaliser.



Les indicateurs



Opération : Procedure IndicateurSimple(req:Tjson; out resp:TJJson);

Qualificatif JSON : KPI

```
var
  json:TJJson;

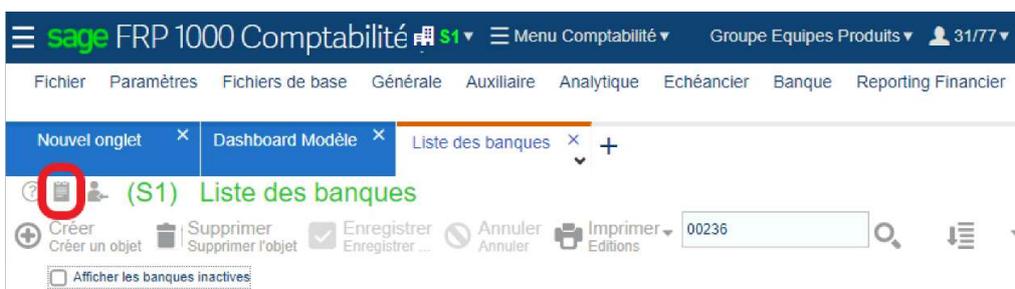
begin

  json := Tjson.Create("");
  resp := json;

  json.result.typeShow := 'column'; // valeurs possibles : "column", "line" ou "square"
  json.result.data[0].displayName := 'Indicateur simple';
  json.result.data[0].value := 1000;
  json.result.data[0].color := 'rgb(0, 220, 0)';
  json.result.data[0].currency := '€'; // Optionnel
  json.result.data[0].subValue := 'Sous-titre'; // Optionnel
  json.result.data[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm"';

end;
```

Les informations précédées de // sont des commentaires
Pour toutes les vignettes, les ouvertures d'écrans (link) contiennent l'URL de l'écran à ouvrir.
L'url est accessible à partir du bouton « Copier l'url de l'écran présent en en-tête des écrans.



Pour la documentation, le même lien a été renseigné, il ouvre l'écran liste des devises.



Opération : Procedure Indicateur2Donnees(req:Tjson; out resp:TJSon);

Qualificatif JSON : KPI

```
var
```

```
  json:TJson;
```

```
begin
```

```
  json := Tjson.Create("");
```

```
  resp := json;
```

```
  json.result.typeShow := 'column'; // valeurs possibles : "column", "line" ou "square"
```

```
  json.result.data[0].displayName := 'Indicateur 2 données';
```

```
  json.result.data[0].value := 1000;
```

```
  json.result.data[0].currency := '€';
```

```
  json.result.data[0].color := 'rgb(51, 92, 109)';
```

```
  json.result.data[0].subValue := 'Sous-titre 1';
```

```
  json.result.data[0].link :=
```

```
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";
```

```
  json.result.data[1].displayName := 'Donnée 2';
```

```
  json.result.data[1].value := 2000;
```

```
  json.result.data[1].currency := '€';
```

```
  json.result.data[1].color := 'rgb(199, 56, 79)';
```

```
  json.result.data[1].subValue := 'Sous-titre 2';
```

```
  json.result.data[1].link := "";
```

```
end;
```



Opération : Procedure Indicateur4Donnees(req:Tjson; out resp:TJJson);

Qualificatif JSON : KPI

```

var
  json:TJJson;

begin

  json := TJJson.Create("");
  resp := json;

  json.result.typeShow := 'square'; // valeurs possibles : "column", "line" ou "square"

  json.result.data[0].displayName := 'Indicateur 4 données';
  json.result.data[0].value := 1000;
  json.result.data[0].currency := '€';
  json.result.data[0].color := 'rgb(0, 220, 0)';
  json.result.data[0].subValue := 'Sous-titre 1';
  json.result.data[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm"';

  json.result.data[1].displayName := 'Donnée 2';
  json.result.data[1].value := 2000;
  json.result.data[1].currency := '€';
  json.result.data[1].color := 'rgb(0, 68, 114)';
  json.result.data[1].subValue := 'Sous-titre 2';
  json.result.data[1].link := "";

  json.result.data[2].displayName := 'Donnée 3';
  json.result.data[2].value := 3000;
  json.result.data[2].currency := '€';
  json.result.data[2].color := 'rgb(0, 115, 194)';
  json.result.data[2].subValue := 'Sous-titre 3';
  json.result.data[2].link := "";

  json.result.data[3].displayName := 'Donnée 4';
  json.result.data[3].value := 4000;
  json.result.data[3].currency := '€';

```

```
json.result.data[3].color := 'rgb(52, 92, 109)';  
json.result.data[3].subValue := 'Sous-titre 4';  
json.result.data[3].link := '';
```

```
end;
```

Les indicateurs de tendances

Choix de l'icône

trendIcon	trendDirection	Résultat
face	plus	
face	equal	
face	moins	
thumb	plus	
thumb	moins	
arrow	plus	
arrow	moins	

Si la propriété « trendIcon » possède la valeur « arrow », il est possible d'inverser les couleurs en renseignant la propriété « trendInverse ».



Opération : Procédure IndicateurTendance(req:Tjson; out resp:TJSon);

Qualificatif JSON : KPI

```

var
  json:TJson;

begin

  json := Tjson.Create("");
  resp := json;

  json.result.typeShow := 'trend';
  json.result.trendIcon := 'arrow'; // valeurs possibles : 'thumb', 'arrow' ou 'face'
  json.result.trendDirection := 'plus'; // valeurs possibles : 'plus', 'equal' ou 'minus'
  //Si la propriété trendIcon possède la valeur arrow, il est possible d'inverser les couleurs en renseignant la propriété trendInverse
  json.result.trendInverse := false;

  json.result.data[0].displayName := 'Indicateur de tendance';
  json.result.data[0].value := '1000,00';
  json.result.data[0].currency := '€';
  json.result.data[0].color := 'rgb(0, 220, 0)';
  json.result.data[0].subValue := 'Sous-titre 1';
  json.result.data[0].footer := 'Information supplémentaire';
  json.result.data[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";

end;

```



Opération : Procedure IndicateurTendance2(req:Tjson; out resp:TJSon);

Qualificatif JSON : KPI

```

var
  json:TJson;

begin

  json := Tjson.Create("");
  resp := json;

  json.result.typeShow := 'trend';
  json.result.trendIcon := 'face'; // valeurs possibles : 'thumb', 'arrow' ou 'face'
  json.result.trendDirection := 'plus'; // valeurs possibles : 'plus', 'equal' ou 'minus'

  json.result.data[0].displayName := 'Indicateur de tendance';
  json.result.data[0].value := 10;
  json.result.data[0].color := 'rgb(153, 173, 182)';
  json.result.data[0].subValue := 'Sous-titre 1';
  json.result.data[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm"';

end;

```



Opération : Procédure IndicateurTendance3(req:Tjson; out resp:TJSon);

Qualificatif JSON : KPI

```
var
```

```
  json:TJson;
```

```
begin
```

```
  json := Tjson.Create("");
```

```
  resp := json;
```

```
  json.result.typeShow := 'trend';
```

```
  json.result.trendIcon := 'thumb'; // valeurs possibles : 'thumb', 'arrow' ou 'face'
```

```
  json.result.trendDirection := 'plus'; // valeurs possibles : 'plus', 'equal' ou 'minus'
```

```
  json.result.data[0].displayName := 'Indicateur de tendance';
```

```
  json.result.data[0].value := 20;
```

```
  json.result.data[0].color := 'rgb(0, 115, 194)';
```

```
  json.result.data[0].subValue := 'Sous-titre 1';
```

```
  json.result.data[0].link :=
```

```
  'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";
```

```
end;
```

Les grilles

Grille avec couleurs

Colonne 1	Colonne 2	Colonne 3	Colonne 4
Ligne 1	21	31	41
Ligne 2	22	32	42
Ligne 3	23	33	43

Opération : Procédure GrilleAvecCouleur(req:Tjson; out resp:TJJson);

Qualificatif JSON : Tabulaire

```
var
  json:TJJson;

begin

  json := Tjson.Create("");
  resp := json;

  // En-têtes de colonnes
  json.result[0].key := 'col1';
  json.result[0].displayName := 'Colonne 1';
  json.result[0].type := 'umlString';
  // Types de données possibles :
  // umlString, umlStringS, umlDateTime, umlDate, Integer,
  // umlQuantity, umlNumeric, umlFloat, umlMoney, umlMoneyDT

  json.result[1].key := 'col2';
  json.result[1].displayName := 'Colonne 2';
  json.result[1].type := 'umlInteger';
  json.result[1].columnStyle := '{"color": "#007bff"}';

  json.result[2].key := 'col3';
  json.result[2].displayName := 'Colonne 3';
  json.result[2].type := 'umlInteger';
  json.result[2].columnStyle := '{"background-color": "rgb(204, 214, 219)"}';
  //Syntaxe à respecter : {"propriété css":"valeur", "autre propriété css":"valeur"}

  json.result[3].key := 'col4';
  json.result[3].displayName := 'Colonne 4';
  json.result[3].type := 'umlInteger';
  json.result[3].columnStyle := '{"color": "#fff", "background-color": "rgb(0, 132, 0)"}';

  // Valeurs
```

```

// Valeurs Colonne 1
json.result[0].values[0] := 'Ligne 1';
json.result[0].values[1] := 'Ligne 2';
json.result[0].values[2] := 'Ligne 3';
// Valeurs Colonne 2
json.result[1].values[0] := 21;
json.result[1].values[1] := 22;
json.result[1].values[2] := 23;
// Valeurs Colonne 3
json.result[2].values[0] := 31;
json.result[2].values[1] := 32;
json.result[2].values[2] := 33;
// Valeurs Colonne 4
json.result[3].values[0] := 41;
json.result[3].values[1] := 42;
json.result[3].values[2] := 43;

// Ouverture d'écran en cliquant sur la colonne 1
json.result[0].link[0] :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";
json.result[0].columnLink[0] := 'col1';

end;

```

Grille avec flip de détail

Colonne 1	Colonne 2	Colonne 3	Colonne Flip
Ligne 1	21	31	2
Ligne 2	22	32	1
Ligne 3	23	33	1

Opération : Procedure GrilleAvecFlip(req:Tjson; out resp:TJson);
Qualificatif JSON : Tabulaire

```

var
  json:TJson;

begin

  json := Tjson.Create("");

```

```

resp := json;

// Colonne 01
json.result[0].key := 'col1'; //end
json.result[0].displayName := 'Colonne 1';
json.result[0].type := 'umlString';
json.result[0].values[0] := 'Ligne 1';
json.result[0].values[1] := 'Ligne 2';
json.result[0].values[2] := 'Ligne 3';
// Ouverture d'écran en cliquant sur la colonne 1
json.result[0].link[0] :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";
json.result[0].columnLink[0] := 'col1';
// Colonne 02
json.result[1].key := 'col2'; //campaign
json.result[1].displayName := 'Colonne 2';
json.result[1].type := 'umlString';
json.result[1].values[0] := '21';
json.result[1].values[1] := '22';
json.result[1].values[2] := '23';
// Colonne 03
json.result[2].key := 'col3'; //society
json.result[2].displayName := 'Colonne 3';
json.result[2].type := 'umlString';
json.result[2].values[0] := '31';
json.result[2].values[1] := '32';
json.result[2].values[2] := '33';
// Colonne 04
json.result[3].key := 'col4'; //todo
json.result[3].displayName := 'Colonne Flip';
json.result[3].type := 'umlInteger';
json.result[3].columnStyle := '{"font-weight": "bold"}';
json.result[3].values[0] := 2;
json.result[3].values[1] := 1;
json.result[3].values[2] := 1;

// 1. Ajout de la colonne
json.result[4].key := 'detail';
json.result[4].displayName := 'Détail';
json.result[4].type := 'object';

// 2. La clé de la colonne qui affichera une liste détaillée au clic est 'col4'.

// 3. Ajout de la liste des en-têtes de colonne à afficher dans la liste détaillée
json.result[4].values[0].col4.columns[0] := 'Détail - Colonne 1';
json.result[4].values[0].col4.columns[1] := 'Détail - Colonne 2';

// 4. Ajout des données à afficher
// L'ordre des données dans la liste doit coïncider avec celui des colonnes.

```

```
// Pour la ligne 0
json.result[4].values[0].col4.values[0].c1 := 'Détail 1 L1 C1';
json.result[4].values[0].col4.values[0].c2 := 'Détail 1 L1 C2';
json.result[4].values[0].col4.values[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";
json.result[4].values[0].col4.values[1].c1 := 'Détail 2 L1 C1';
json.result[4].values[0].col4.values[1].c2 := 'Détail 2 L1 C2';
json.result[4].values[0].col4.values[1].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";

// Pour la ligne 1
json.result[4].values[1].col4.values[0].c1 := 'Détail 1 L2 C1';
json.result[4].values[1].col4.values[0].c2 := 'Détail 1 L2 C2';
json.result[4].values[1].col4.values[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";

// pour la ligne 2
json.result[4].values[2].col4.values[0].c1 := 'Détail 1 L3 C1';
json.result[4].values[2].col4.values[0].c2 := 'Détail 1 L3 C2';
json.result[4].values[2].col4.values[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";

end;
```

Les listes de valeurs

Liste de valeurs	
Ventes	14,700.00
Commandes	4,200.00
Factures	13,500.00
dont Avoirs	-3,000.00
Achats	-20,000.00
Total	-5,300.00

L'utilisation de la vignette « Liste de valeurs » n'est pas recommandée lorsqu'il y a beaucoup d'éléments car le scroll n'est pas présent sur la vignette.

Il est possible d'ajouter des indentations (deux niveaux d'indentation possibles) pour regrouper certaines lignes d'éléments « outstandings » :

- l'élément « outstandings » précédent les lignes indentées doit avoir la propriété « indent » à 0
- les éléments « outstandings » indentés de premier niveau doivent avoir la propriété « indent » à 1
- les éléments « outstandings » indentés de second niveau doivent avoir la propriété « indent » à 2

Par défaut, la vignette ajoute deux décimales aux valeurs numériques.

Pour modifier le nombre de décimales, il faut utiliser la propriété « decimal ».

La couleur de la police des titres passe en bleu lorsqu'une Url est renseignée.

Opération : Procedure ListeDeValeurs(req:Tjson; var resp:TObject);

Qualificatif JSON : Liste

```
// la liste outstandings peut contenir 0..* éléments
```

```
// la liste subtotal peut contenir 0..* éléments
```

```
// l'élément total peut contenir 0..1 élément
```

```
var
```

```
  json:TJson;
```

```
begin
```

```
  json := Tjson.Create("");
```

```
  resp := json;
```

```
// Parent ident = 0
json.result.outstandings[0].label := 'Ventes';
json.result.outstandings[0].value := 14700;
json.result.outstandings[0].decimal := 2;
json.result.outstandings[0].indent := 0;
json.result.outstandings[0].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";
//
// Niveau 1 ident = 1
json.result.outstandings[1].label := 'Commandes';
json.result.outstandings[1].value := 4200;
json.result.outstandings[1].decimal := 2;
json.result.outstandings[1].indent := 1;
//
json.result.outstandings[2].label := 'Factures';
json.result.outstandings[2].value := 13500;
json.result.outstandings[2].decimal := 2;
json.result.outstandings[2].indent := 1;
//
// Niveau 2 ident = 3
json.result.outstandings[3].label := 'dont Avoirs';
json.result.outstandings[3].value := -3000;
json.result.outstandings[3].decimal := 2;
json.result.outstandings[3].indent := 2;

// Parent ident = 0
json.result.outstandings[4].label := 'Achats';
json.result.outstandings[4].value := -20000;
json.result.outstandings[4].decimal := 2;
json.result.outstandings[4].indent := 0;
json.result.outstandings[4].link :=
'frm://egescom_listdevises.dfm/AADD387B1DAC03FF0003000000E04BE/?params=""&options="seoRootForm";

json.result.total.label := 'Total';
json.result.total.value := -5300.00;

end;
```

Les listes simples



Ce type de vignette permet de créer des menus ou raccourcis vers de écrans de la Ligne 1000.

Dès lors qu'une Url d'écran est paramétrée, la police des titres passe en bleu.

Opération : Procédure ListeSimple(req:Tjson; var resp:TObject);

Qualificatif JSON : Liste

```
var
  json:TJson;
begin
  json := Tjson.Create("");
  resp := json;

  json.result.outstandings[0].label := 'Général';
  json.result.outstandings[0].indent := 0;

  json.result.outstandings[1].label := 'Interrogation de compte';
  json.result.outstandings[1].indent := 1;
  json.result.outstandings[1].link :=
'frm://interrogationcomptegeneral.dfm//?seolInsert=Yes&params="ArgCompte=mptString(10100000)&ArgDateInf=mptDateTime(01/
01/2021)&ArgDateSup=mptDateTime(31/12/2021)&ArgPTypeLot=mptString(Réel)"&options="seoRootForm"';

  json.result.outstandings[2].label := 'Consultation des pièces';
  json.result.outstandings[2].indent := 1;
  json.result.outstandings[2].link := 'frm://ecompta_consultationpiece.dfm//?seolInsert=Yes&params=""&options="seoRootForm"';

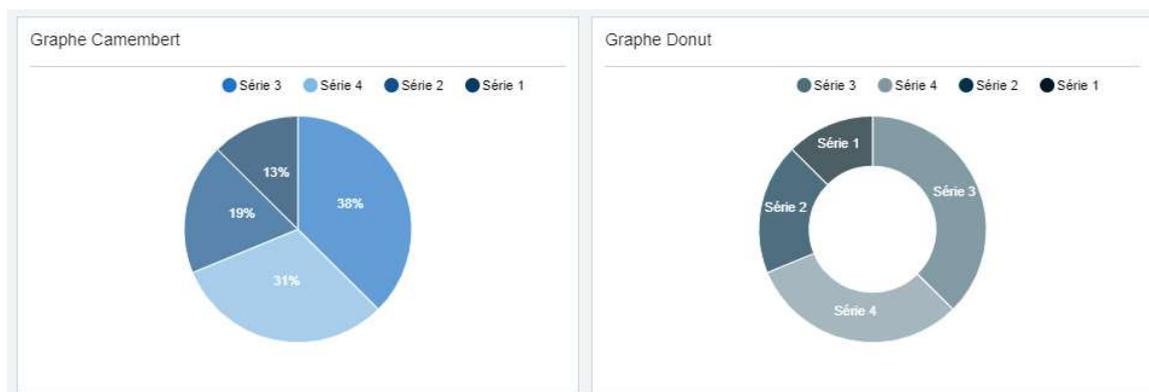
  json.result.outstandings[3].label := 'Auxiliaire';
  json.result.outstandings[3].indent := 0;
```

```
json.result.outstandings[4].label := 'Interrogation de comptes auxiliaires';
json.result.outstandings[4].indent := 1;
json.result.outstandings[4].link :=
'frm://ecompta_interrogationcomptetiers.dfm/AADD387B1DAC03FF078B000004840001/?params="ArgDateInf=mptDateTime(01/01
/2021)&ArgDateSup=mptDateTime(31/12/2021)&ProcTypeLot=mptInteger(1)"&options="seoRootForm";

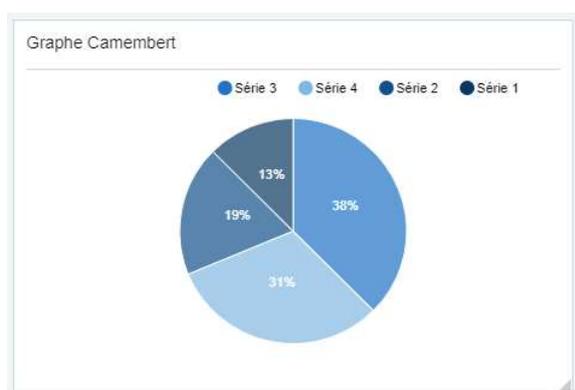
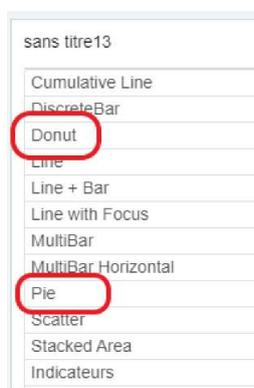
json.result.outstandings[5].label := 'Balance auxiliaire';
json.result.outstandings[5].indent := 1;
json.result.outstandings[5].link :=
'frm://lrbalanceauxsimple.dfm//?seoInsert=Yes&params="ArgAuxiliaire=mptBoolean(true)&ArgUtiliserDevise=mptBoolean(false)"&
options="seoRootForm";

end;
```

Les graphiques « Camembert » ou « Donut »



Les deux graphiques utilisent un code identique, lors de l'ajout dans la vignette, il convient de sélectionner « Pie » ou « Donut ».



Opération : Procédure GrapheCamembert(req:Tjson; var resp:TObject);

Qualificatif JSON : Chart

// Type de données possible : umlString, umlStringS, umlDateTime, umlDate
// umlInteger, umlQuantity, umlNumeric, umlFloat, umlMoney, umlMoneyDT

```

// x : libellé de la série
// xtype : type de donnée de la série
// y : valeur de la série
// ytype : type de donnée de la valeur
// ycurrency : symbole monétaire s'affichant après la valeur
// color : couleur du secteur

var
  json:TJson;

begin
  json := Tjson.Create("");
  resp := json;

  json.result.data[0].values[0].x := 'Série 1';
  json.result.data[0].values[0].xtype := 'umlString';
  json.result.data[0].values[0].y := 10;
  json.result.data[0].values[0].ytype := 'umlInteger';
  json.result.data[0].values[0].color := 'rgb(0, 59, 99)';

  json.result.data[0].values[1].x := 'Série 2';
  json.result.data[0].values[1].xtype := 'umlString';
  json.result.data[0].values[1].y := 15;
  json.result.data[0].values[1].ytype := 'umlInteger';
  json.result.data[0].values[1].color := 'rgb(0, 83, 139)';

  json.result.data[0].values[2].x := 'Série 3';
  json.result.data[0].values[2].xtype := 'umlString';
  json.result.data[0].values[2].y := 30;
  json.result.data[0].values[2].ytype := 'umlInteger';
  json.result.data[0].values[2].color := 'rgb(0, 119, 200)';

  json.result.data[0].values[3].x := 'Série 4';
  json.result.data[0].values[3].xtype := 'umlString';
  json.result.data[0].values[3].y := 25;
  json.result.data[0].values[3].ytype := 'umlInteger';
  json.result.data[0].values[3].color := 'rgb(128, 187, 227)';

end;

```



Opération : Procedure GrapheDonut(req:Tjson; var resp:TObject);

Qualificatif JSON : Chart

```

var
  json:TJson;

begin
  json := Tjson.Create("");
  resp := json;

  json.result.data[0].values[0].x := 'Série 1';
  json.result.data[0].values[0].xtype := 'umlString';
  json.result.data[0].values[0].y := 10;
  json.result.data[0].values[0].ytype := 'umlInteger';
  json.result.data[0].values[0].color := 'rgb(0, 25, 36)';

  json.result.data[0].values[1].x := 'Série 2';
  json.result.data[0].values[1].xtype := 'umlString';
  json.result.data[0].values[1].y := 15;
  json.result.data[0].values[1].ytype := 'umlInteger';
  json.result.data[0].values[1].color := 'rgb(0, 51, 73)';

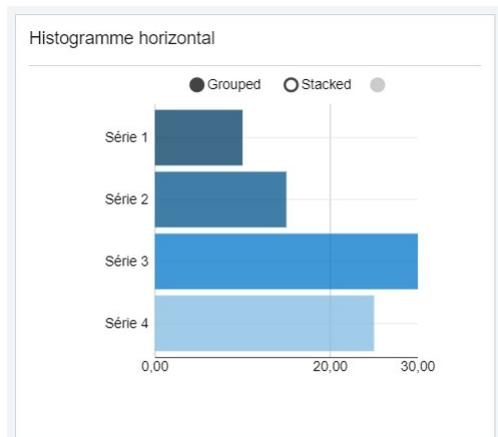
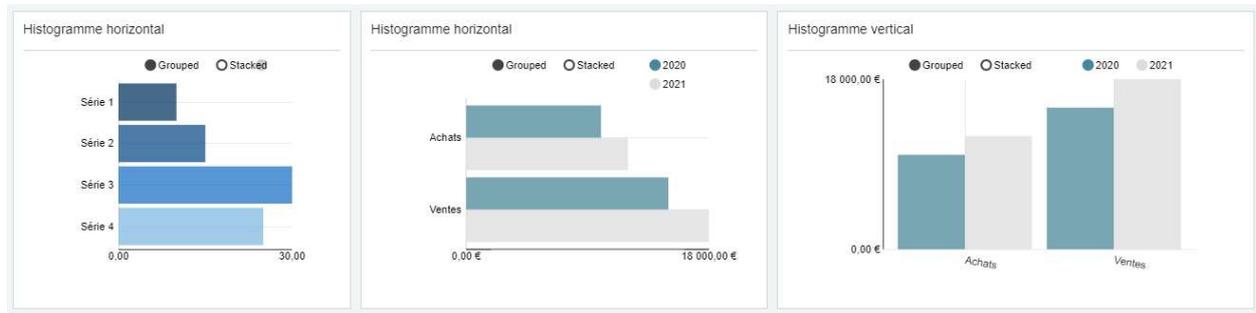
  json.result.data[0].values[2].x := 'Série 3';
  json.result.data[0].values[2].xtype := 'umlString';
  json.result.data[0].values[2].y := 30;
  json.result.data[0].values[2].ytype := 'umlInteger';
  json.result.data[0].values[2].color := 'rgb(77, 112, 127)';

  json.result.data[0].values[3].x := 'Série 4';
  json.result.data[0].values[3].xtype := 'umlString';
  json.result.data[0].values[3].y := 25;
  json.result.data[0].values[3].ytype := 'umlInteger';
  json.result.data[0].values[3].color := 'rgb(128, 153, 164)';

end;

```

Les graphiques « Histogramme »



Opération : Procedure GrapheCamembert(req:Tjson; var resp:TObject);

Qualificatif JSON : Chart

```
var
  json:TJson;

begin
  json := Tjson.Create("");
  resp := json;

  json.result.data[0].values[0].x := 'Série 1';
  json.result.data[0].values[0].xtype := 'umlString';
  json.result.data[0].values[0].y := 10;
  json.result.data[0].values[0].ytype := 'umlInteger';
  json.result.data[0].values[0].color := 'rgb(0, 59, 99)';

  json.result.data[0].values[1].x := 'Série 2';
  json.result.data[0].values[1].xtype := 'umlString';
  json.result.data[0].values[1].y := 15;
  json.result.data[0].values[1].ytype := 'umlInteger';
```

```
json.result.data[0].values[1].color := 'rgb(0, 83, 139)';
```

```
json.result.data[0].values[2].x := 'Série 3';  
json.result.data[0].values[2].xtype := 'umlString';  
json.result.data[0].values[2].y := 30;  
json.result.data[0].values[2].ytype := 'umlInteger';  
json.result.data[0].values[2].color := 'rgb(0, 119, 200)';
```

```
json.result.data[0].values[3].x := 'Série 4';  
json.result.data[0].values[3].xtype := 'umlString';  
json.result.data[0].values[3].y := 25;  
json.result.data[0].values[3].ytype := 'umlInteger';  
json.result.data[0].values[3].color := 'rgb(128, 187, 227)';
```

```
end;
```



Opération : Procedure GrapheHistogramme(req:Tjson; var resp:TObject);

Qualificatif JSON : Chart

```
var  
  json:TJson;  
  
begin  
  json := Tjson.Create("");  
  resp := json;  
  
  json.result.data[0].key := '2020';  
  json.result.data[0].color := 'rgb(73,137,153)';  
  
  json.result.data[0].values[0].x := 'Achats';
```

```
json.result.data[0].values[0].xtype := 'umlString';
json.result.data[0].values[0].y := 10000;
json.result.data[0].values[0].ytype := 'umlNumeric';
json.result.data[0].values[0].ycurrency := '€';
json.result.data[0].values[0].color := 'rgb(73,137,153)';

json.result.data[0].values[1].x := 'Ventes';
json.result.data[0].values[1].xtype := 'umlString';
json.result.data[0].values[1].y := 15000;
json.result.data[0].values[1].ytype := 'umlNumeric';
json.result.data[0].values[1].ycurrency := '€';
json.result.data[0].values[1].color := 'rgb(73,137,153)';

json.result.data[1].key := '2021';
json.result.data[1].color := 'rgb(222,222,222)';

json.result.data[1].values[0].x := 'Achats';
json.result.data[1].values[0].xtype := 'umlString';
json.result.data[1].values[0].y := 12000;
json.result.data[1].values[0].ytype := 'umlNumeric';
json.result.data[1].values[0].ycurrency := '€';
json.result.data[1].values[0].color := 'rgb(222,222,222)';

json.result.data[1].values[1].x := 'Ventes';
json.result.data[1].values[1].xtype := 'umlString';
json.result.data[1].values[1].y := 18000;
json.result.data[1].values[1].ytype := 'umlNumeric';
json.result.data[1].values[1].ycurrency := '€';
json.result.data[1].values[1].color := 'rgb(222,222,222)';

end;
```

Les filtres

Filtre global

Un filtre global est un composant visuel permettant de filtrer les données des vignettes d'un tableau de bord sur un critère donné.

Un tableau de bord peut proposer plusieurs filtres globaux. Lors de l'application d'un filtre, toutes les vignettes sont réactualisées.

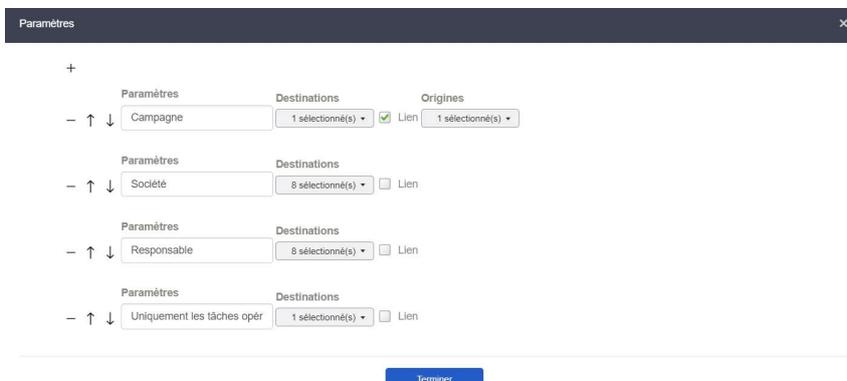


Ajout d'un filtre global

Au niveau du concepteur de tableau de bord et sur le tableau de bord concerné, il est nécessaire d'aller sur les paramètres du tableau de bord.

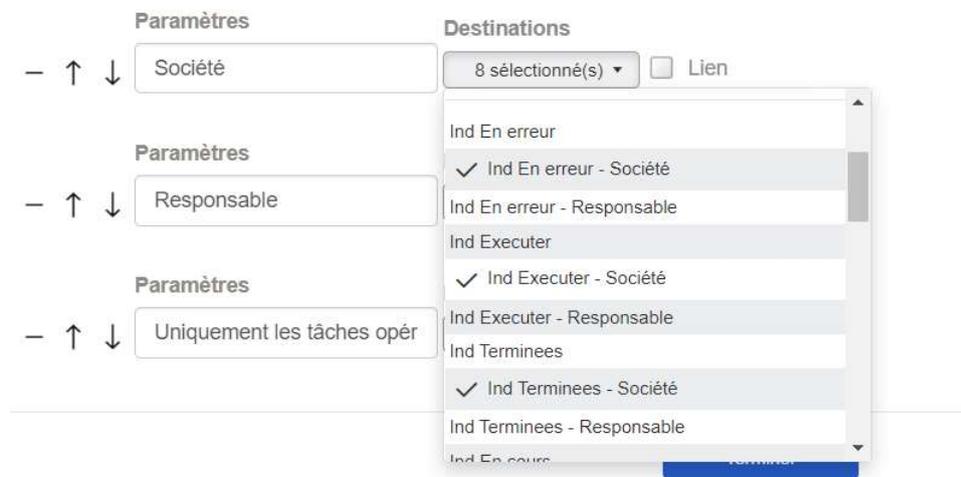


et ajouter un filtre :



Au niveau de Paramètres, saisir un libellé, ce libellé apparaîtra sur le tableau de bord ; et sélectionner les destinations. c'est-à-dire les colonnes des vignettes dont les valeurs serviront au filtre.

Dans l'exemple ci-dessous, un filtre sur les sociétés est créé. Ce filtre proposera les valeurs issues de la colonne « Société ».



Utilisation du filtre dans les méthodes de service

Lors de l'application d'un filtre, toutes les vignettes sont réactualisées.

Dans le cas d'une vignette créée par une méthode de service, une requête est envoyée vers le serveur avec un paramètre correspondant au(x) filtre(s) sélectionné(s) :

```
{"glbFilter": "([code de la vignette]=[valeur sélectionnée])"}
```

Exemple de paramètre de requête avec deux filtres sélectionnés :

```
{"glbFilter": "(society='S4') AND (campaign='Campagne 02')"}
```

Exemple de paramètre de requête avec un filtre multiple sélectionné :

```
{"glbFilter": "(campaign='Campagne 02' OR campaign='Campagne 03')"}
```

Récupération du paramètre dans une méthode de service

Il est nécessaire :

- d'utiliser `req.Exists('glbFilter')` pour tester l'existence du paramètre
- d'utiliser `VarToStr(req.glbFilter)` pour récupérer le filtre

Type

```
Campagne = viewOf(TCampagneOperation)
CodeSociete : string = TiersSociete.code;
CodeModele : string = ModeleCampagne.code;
IDCampagne : TOID = oid;
LibelleCampagne : string = Caption;
DateFinCampagne : datetime = dateFinPrevue;
Statut : integer = statut;
Responsable : string = Responsable.code;
end;
```

var

```
json : TJson;
vSel : TSelector;
vCampagne : Campagne;
vWhere : string;
```

begin

```
vWhere := "";
json := Tjson.Create("");
resp := json;
```

```
// est-ce que le paramètre "glbFilter" existe ?
```

```
if req.Exists('glbFilter') then
```

```
// récupération de son contenu pour filtrer les données
```

```
TFacadeOutils._AjoutExpression(vWhere, 'and', VarToStr(req.glbFilter));
```

```
vSel := Campagne.CreateSelector(vWhere, 'CodeSociete, -DateFinCampagne', false, []);
```

```
// ...
```

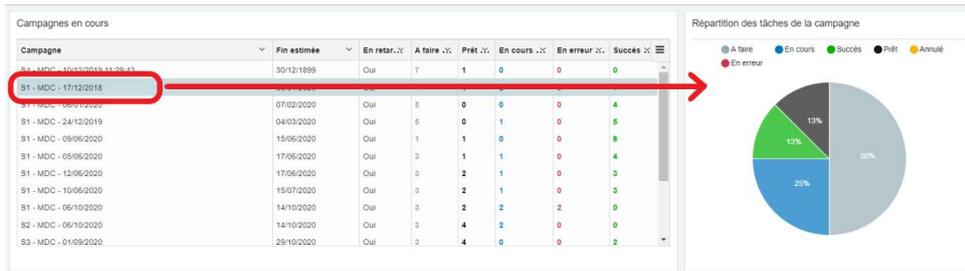
```
end;
```

_filtre entre vignettes

Un filtre entre vignettes permet de filtrer les données d'un groupe de vignettes par rapport à un critère issu d'une autre vignette du tableau de bord.

Un tableau de bord peut proposer plusieurs filtres entre vignettes.

Par exemple dans le tableau de bord des campagnes, dès lors que l'on clique sur le libellé d'une campagne en cours, le graphique affiche la répartition des tâches par statut.

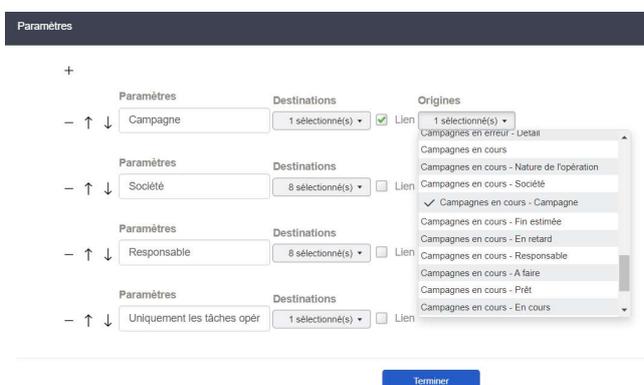


Lors de l'application d'un filtre, seules les vignettes « Destinations » de ce filtre sont réactualisées.

Ajout d'un filtre entre vignettes

Au niveau de Paramètres, ajouter un filtre :

- saisir le libellé, ce libellé n'apparaîtra pas sur le tableau de bord
- sélectionner les destinations. c'est-à-dire les colonnes des vignettes dont les valeurs serviront au filtre.
- cliquer sur la coche « Lien » pour afficher la zone « Origines »
- sélectionner les « Origines », c'est-à-dire les colonnes des vignettes dont les valeurs serviront au filtre.



Utilisation du filtre dans les méthodes de service

Description du paramètre de requête

Lors de l'application d'un filtre, les vignettes sélectionnées dans « Destinations » sont réactualisées.

Dans le cas d'une vignette créée par une méthode de service, une requête est envoyée vers le serveur avec un paramètre correspondant au(x) filtre(s) sélectionné(s) :

```
{"filter":{"[code]":"[valeur sélectionnée]"}}
```

Exemple de paramètre de requête :

```
{"filter":{"campaign":"Campagne 01"}}
```

Exemple de paramètre de requête avec deux filtres sélectionnés :

```
{"filter":{"campaign":"Campagne 01","society":"S1"}}
```

Récupération du paramètre dans une méthode de service

Il est nécessaire :

- d'utiliser `req.Exists('filter')` et `req.Exists('filter.unCode')` pour tester l'existence du paramètre ;
- d'utiliser `req.filter.unCode` pour récupérer la valeur du filtre dont le nom est `unCode`.

Contrairement à un filtre global, le filtre entre vignettes ne permet de récupérer que la valeur à filtrer. Le code du serveur doit connaître le nom du filtre pour récupérer sa valeur.

Exemple de code avec un filtre global et un filtre entre vignettes :

Type

```
Tache = viewOf(TTacheCampagne)
CodeSociete : string = EtapeCampagne.CampagneOperation.TiersSociete.code;
IDSociete : string = EtapeCampagne.CampagneOperation.oidTiersSociete;
LibelleCampagne : string = EtapeCampagne.CampagneOperation.Caption;
IDCampagne : string = EtapeCampagne.oidCampagneOperation;
DateFinCampagne : datetime = EtapeCampagne.CampagneOperation.dateFinPrevue;
LibelleEtapeCourante : string = EtapeCampagne.CampagneOperation.EtapeCourante.EtapeModele.Caption;
StatutCampagne : integer = EtapeCampagne.CampagneOperation.statut;
Responsable : string = EtapeCampagne.CampagneOperation.Responsable.Caption;
Statut : integer = dbfActivityStatus.StatusType;
StatutLibelle : string = dbfActivityStatus.Caption;
DatePrevue : datetime = DueDate;
LibelleEtape : string = EtapeCampagne.EtapeModele.Caption;
OrdreEtape : integer = EtapeCampagne.EtapeModele.ordre;
Operateur : string = Operateur.Caption;
IDTache : string = oid;
LibelleTache : string = OperationCampagne+Caption;
end;

var
  json : TJson;
  vSel : TSelector;
  vWhere : string;
  Args : Array of variant;
begin
  json := Tjson.Create("");
  resp := json;
  vWhere := '(StatutCampagne <> %1) and (StatutCampagne <> %2) and ((Statut = %3) or (Statut = %4))';

  // Filtre global
  if req.Exists('glbFilter') then
    TFacadeOutils._AjoutExpression(vWhere, 'and', VarToStr(req.glbFilter));

  // Filtre entre vignettes
  if (req.Exists('filter')) then
    begin
      // est-ce que le filtre "IDCampagne" existe ?
      if (req.Exists('filter.IDCampagne')) then
        begin
          TFacadeOutils._AjoutExpression(vWhere, 'and', 'IDCampagne = %5');
          Args[4] := req.filter.IDCampagne; // récupération de la valeur du filtre
        end;
    end;

  Args[0] := StatutTache_Annule;
  Args[1] := StatutTache_Termine;
```

```
Args[2] := cstActivity_Ready;  
Args[3] := cstActivity_InProgress;
```

```
vSel := Tache.CreateSelector(vWhere, '-DateFinCampagne, CodeSociete, IdCampagne, OrdreEtape, Statut', false, Args);
```

```
// ...
```

```
end;
```



www.sage.com

© 2020 The Sage Group plc or its licensors. All rights reserved. Sage, Sage logos, and Sage product and service names mentioned herein are the trademarks of The Sage Group plc or its licensors. All other trademarks are the property of their respective owners.