**HOW TO GUIDES**

**Abdelhak SALLAOUI**

**Construction des actionS chart**

Table des matières

[01. Objectif 3](#_Toc9608705)

[02. Exemple 4](#_Toc9608706)

[03. Contruction du graphique et analyse du comportement 5](#_Toc9608707)

[04. Réception de données et application de filtre 7](#_Toc9608708)

# Objectif

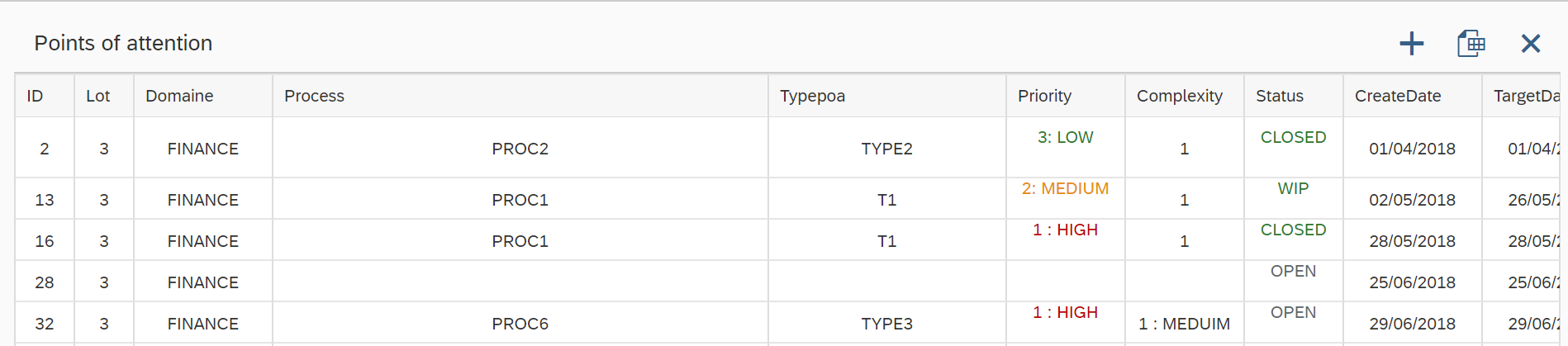
Souvent sur les projets, les utilisateurs ne se satisfont plus des reports classiques avec lignes, colonnes et leurs filtres et tris. Ils réclament de plus en plus d’outils graphiques qui permettent une navigation de la vue globale à la vue détaillées.

L’idée ici est de réaliser des smart graphs permettant une navigation vers d’autres vues xml en appliquant un filtre sur la partie sélectionnée, ceci permettra aux utilisateurs une belle expérience Fiori.

# Exemple

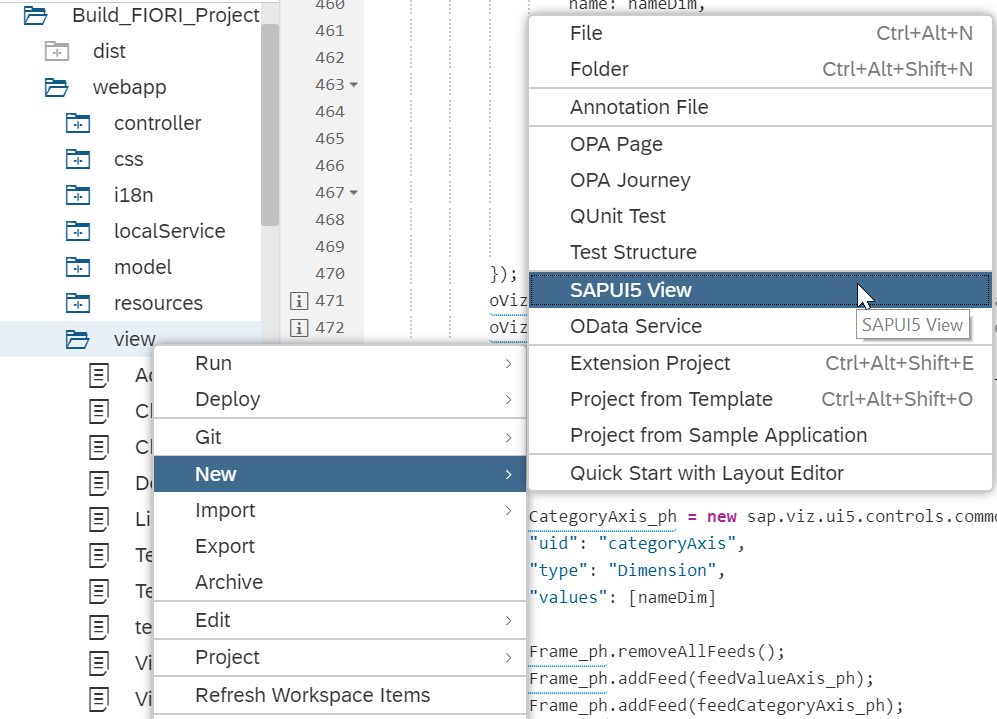
# 

Dans la figure ci-dessus, j’ai cliqué sur la 5ème case de la 3ème colonne, ceci va nous amener vers une 2ème vue XML, en affichant que les POA liés au lot 3 et au domaine Finance.



# contruction du graphique et analyse du comportement

Pour ce faire, il va falloir dans un premier temps créer une application Fiori from scratch (SAPUI5 Application), ensuite créer une nouvelle vue comme suit :



**Remarque :** Ceci va nous permettre de créer la vue et le contrôleur à la fois.

Le 2ème pas consiste à rajouter des balises dans la vue XML liée à notre contrôleur, qui représente le corps de notre graphe.





Et comme nous voulons lui ajouter une action, il va falloir rajouter un paramètre « selectData ». Ce paramètre va nous lier directement avec le contrôleur associé à cette vue XML.

**Coté Contrôleur**

Dans la partie contrôleur, nous devons ajouter une fonction paramétrée :

Remarque : la fonction doit être nommée de la même façon que le paramètre coté vue XML.



Le paramètre oEvent va nous donner une idée du comportement de l’utilisateur (sans ce param nous ne pouvons pas savoir s’il a appuyé sur toute la colonne (Lot) ou sur un bout de la colonne (Domaine)).

Pour analyser le comportement de l’utilisateur, nous pouvons suivre cet algorithme :

* Si la longueur retournée par la méthode getParameter("data") du paramètre oEvent est inférieure ou égale à 1 (Le user a sélectionné un domaine), on procède dans ce cas à la recherche du domaine sélectionné et le lot sélectionné.
* Si non, si la longueur retournée par la méthode getParameter("data") du paramètre oEvent est supérieur à 1 (Le user a sélectionné un lot) on procède dans ce cas à la recherche du lot sélectionné seulement.
* Une fois les informations collectées, nous pouvons par la suite naviguer vers la page qui appliquera le filtre.

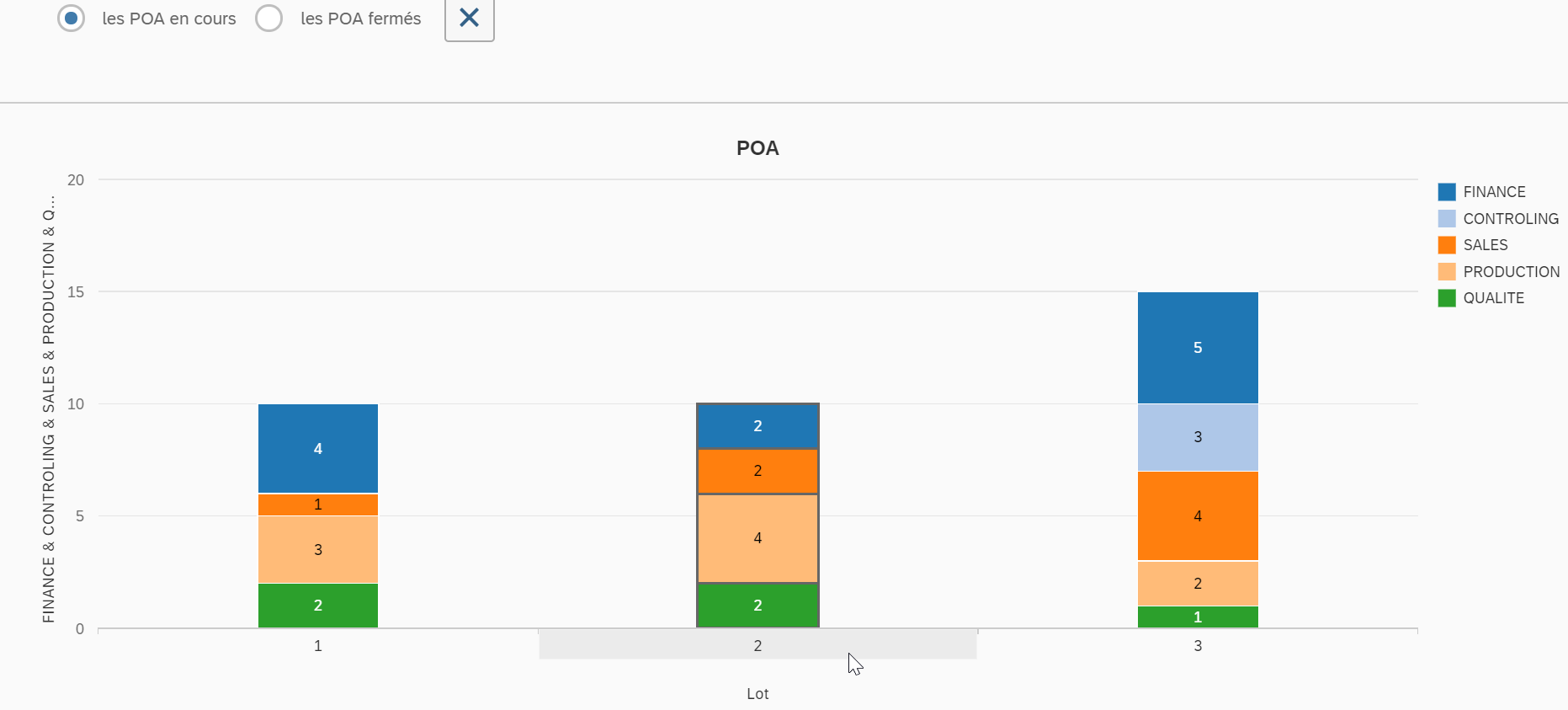
# Réception de données et application de filtre

Le contrôleur émetteur doit importer les données en se basant sur les getters, et appliquer par la suite un filtre (éventuellement deux filtres - tout dépends de ce que l’utilisateur veut faire-).

La fonction coté contrôleur émetteur doit d’abord récupérer le Binding existant de la smart table et le mettre dans un objet, par la suite construire un objet Filter et finalement appliquer le filtre via la méthode filter() de l’objet Binding.



# 2eme cas en revenant à l’utilisateur



Dans cet exemple nous avons cliqué sur la 2ème colonne entièrement pour sélectionner tout le lot numéro 2, ceci va déclencher la fonction coté contrôleur nommée « myOnClickHandler » cette dernière va faire une analyse du comportement de l’utilisateur (en se basant sur l’algorithme mentionné dessus), et permettre une navigation vers la 2ème vue.

Le contrôleur émetteur va faire d’un côté une récolte de l’information et de l’autre applique le filtre sur les données pour n’avoir que les lignes qui correspondent au lot 2. Enfin nous aurons ce résultat :

