# Créer un dashboard

## Objectif :



Dans cet exemple très basique on crée une application contenant un dashboard. Celui-ci comprend 2 ***widgets***. Quand on clique sur l’un d’eux il s’ouvre (comportement par défaut):



## Mise en place :

<!DOCTYPE html>

<html>

<head>

<title>**Dashboard**</title>

<meta charset=**"UTF-8"**/>

<link href=**"css/jsa.css"** rel=**"stylesheet"** media=**"all"** type=**"text/css"**>

<link href=**"css/multiDisplay.css"** rel=**"stylesheet"** media=**"all"** type=**"text/css"**>

<link href=**"css/editor2.css"** rel=**"stylesheet"** media=**"all"** type=**"text/css"**>

<link href=**"img/favicon.png"** rel=**"icon"** media=**"all"** type=**"text/css"**>

<meta name=**"viewport"** content=**"width=450, user-scalable=no"**>

<meta charset=**"utf-8"**>

<style>

</style>

</head>

<body>

<script src=**"js/jsa.js"**></script>

<script src=**"js/jsa.cls.js"**></script>

<script src=**"js/jsa.appliMobile.js"**></script>

<script src=**"js/multiDisplay.js"**></script>

<script src=**"js/dashboard.js"**></script>

<script src=**"js/grid.js"**></script>

<script src=**"js/selector.js"**></script>

<script src=**"js/editor2.js"**></script>

<script>

</script>

<script>

jsa.appliMobile**({**

 id**:**"myOrganization"**,**

 label**:**"My organization"**,**

 objects**:[],**

 displayBody**:*function*(){**

 ***return*** **{**

 tag**:**"div"**,**

 className**:**"appliMobile\_body"**,**

 owner**:*this*,**

 isElt**:**"\_divBody"**,**

 children**:[{**

 tag**:**"dashboard"**,**

 label**:**"Mon organisation"**,**

 labelsProperty**:**"myLabel"**,**

 childrenDisplayIds**:[**

 **{**

 id**:**"test1"**,**

 expand**:**"expand"**,**

 display**:**"grid"**,**

 x**:**10**,**

 y**:**40**,**

 **},**

 **{**

 id**:**"test2"**,**

 expand**:**"expand"**,**

 display**:**"grid"**,**

 x**:**100**,**

 y**:**40**,**

 **}**

 **],**

 records**:[**

 **{**

 tag**:**"info1"**,**

 id**:**"test1"**,**

 myLabel**:**"Info 1"**,**

 records**:[**

 **{**id**:**1**,** label**:**"item 1"**,** prop1**:**"aaa1"**,** prop2**:**"bbb1"**,** prop3**:**"ccc1"**,** prop4**:**"ddd1"**},**

 **{**id**:**2**,** label**:**"item 2"**,** prop1**:**"aaa2"**,** prop2**:**"bbb2"**},**

 **{**id**:**3**,** label**:**"item 3"**,** prop1**:**"aaa3"**,** prop2**:**"bbb3"**},**

 **{**id**:**4**,** label**:**"item 4"**,** prop1**:**"aaa4"**,** prop2**:**"bbb4"**},**

 **{**id**:**5**,** label**:**"item 5"**,** prop1**:**"aaa5"**,** prop2**:**"bbb5"**},**

 **{**id**:**6**,** label**:**"item 6"**,** prop1**:**"aaa6"**,** prop2**:**"bbb6"**},**

 **{**id**:**7**,** label**:**"item 7"**,** prop1**:**"aaa7"**,** prop2**:**"bbb7"**},**

 **{**id**:**8**,** label**:**"item 8"**,** prop1**:**"aaa8"**,** prop2**:**"bbb8"**},**

 **{**id**:**9**,** label**:**"item 9"**,** prop1**:**"aaa9"**,** prop2**:**"bbb9"**},**

 **{**id**:**10**,** label**:**"item 10"**,** prop1**:**"aaa10"**,** prop2**:**"bbb10"**}**

 **]**

 **},**

 **{**

 tag**:**"info2"**,**

 id**:**"test2"**,**

 myLabel**:**"Info 2"**,**

 records**:[**

 **{**id**:**1**,** label**:**"item 1"**,** prop1**:**"aaa1"**,** prop2**:**"bbb1"**,** prop3**:**"ccc1"**,** prop4**:**"ddd1"**},**

 **{**id**:**2**,** label**:**"item 2"**,** prop1**:**"aaa2"**,** prop2**:**"bbb2"**},**

 **{**id**:**3**,** label**:**"item 3"**,** prop1**:**"aaa3"**,** prop2**:**"bbb3"**},**

 **{**id**:**4**,** label**:**"item 4"**,** prop1**:**"aaa4"**,** prop2**:**"bbb4"**},**

 **{**id**:**5**,** label**:**"item 5"**,** prop1**:**"aaa5"**,** prop2**:**"bbb5"**},**

 **{**id**:**6**,** label**:**"item 6"**,** prop1**:**"aaa6"**,** prop2**:**"bbb6"**},**

 **{**id**:**7**,** label**:**"item 7"**,** prop1**:**"aaa7"**,** prop2**:**"bbb7"**},**

 **{**id**:**8**,** label**:**"item 8"**,** prop1**:**"aaa8"**,** prop2**:**"bbb8"**},**

 **{**id**:**9**,** label**:**"item 9"**,** prop1**:**"aaa9"**,** prop2**:**"bbb9"**},**

 **{**id**:**10**,** label**:**"item 10"**,** prop1**:**"aaa10"**,** prop2**:**"bbb10"**}**

 **]**

 **}**

 **]**

 **}**

 **]**

 **};**

 **}**

 **});**

appli.mobile**=*true*;**

appli.smallScreen**=*true*;**

</script>

</body>

</html>

Dans cet exemple chaque enregistrement (record) doit avoir une propriété identifiante (toujours ***id***).

La façon doit il s’affichera sera donné par la propriété ***childrenDisplayIds***. C’est un tableau dans lequel chaque objet *childDisplay* correspond à un ensemble de **propriétés d’affichage**. Le bon objet est associé grâce à sa propriété ***id.*** On peut utiliser aussi se passer sur le ***tag*** des enregistrements au lieu de leur identifiant au moyen de la propriété ***childrenDisplayTags*** : elle fonctionne de la même façon sauf que la propriété *id* de chaque childDisplay doit correspondre à un *tag*.

## Structure

{

tag:"dashboard",

label:"Mon organisation",

labelsProperty:"myLabel", // propriété commune des enregistrements qui servira au label

childrenDisplayIds:[{childDisplay}], // propriétés d’affichage de chaque enregistrement en fonction de son id

childrenDisplayTags:[{childDisplay}], // propriétés d’affichage de chaque enregistrement en fonction de son tag

records:[{record}] // enregistrements à afficher

}

Où childDisplay = {

 id:"test1", // identifiant correspondant à un enregistrement

 expand:"expand", // mode d’ouverture du widget

 display:"grid", // type de widget pour afficher l’enregistrement une fois ouvert

 x:10, // position x du widget (pas encore ouvert)

 y:40, // position x du widget (pas encore ouvert)

}

## Affichage des widgets à l’état réduit

### Label

Par défaut les widgets sont à l’état réduits. Les widget apparaissent en général avec un **label** :



Ce label doit être présent dans l’une des **propriétés** de **chaque enregistrement**. Le nom de cette propriété peut être définit à 2 endroits différents :

* Au niveau du dashboard :(propriété ***labelsProperty***)



Cela est valable si tous les enregistrements ont cette même propriété

* Au niveau des *childDisplay* (propriété ***iconsProperty***):



### Icone

On peut aussi définir une **icône** :



On peut définit le nom de la propriété indiquant l’icône comme précédemment :

* Au niveau du dashboard :(propriété ***iconsProperty***)
* Au niveau des *childDisplay* (propriété ***iconProperty***):

On peut aussi définir une icône générique. On peut ainsi définir une icône par défaut pour chaque *childDisplay* :





### Commentaire

Il est possible d’ajouter un commentaire :



On le définit au niveau des *childDisplay* :



### Alignement

Il est possible de définir l’alignement au moyen de la propriété ***direction***. Définie au niveau des *childDisplay*, Elle a 2 valeurs possibles :

* ***horizontal*** (widget de gauche ci-dessous)
* ***vertical*** (widget de droite ci-dessous, valeur par défaut)



## Types d’affichage des widgets une fois ouverts

On définit le type de widget de l’enregistrement utilise la propriété d’affichage ***display.***

Les valeurs possibles sont :

dashboard :

L’enregistrement s’affiche sous la forme d’un dashboard (inclus dans celui-ci !)

list :

(nécessite d’inclure ***list.js***)



 

### cards

(nécessite ***cards.js***)



### grid

(nécessite ***grid.js***)



### form

(nécessite ***form2.js***)



### properties

(nécessite ***properties.js***, ***formInput.js*** et ***jsa.application.js***)



### obj

Affiche l’enregistrement au moyen de la méthode ***display***() d’un **object** enregistré par *registerClass*(). L’association entre l’enregistrement et l’objet se fait par le **tag** de l’enregistrement.



jsa.registerClass({

 tag:"info1",

 inherits:"graphic",

 proto:{

 beforeInit:function(){

 },

 display:function(){

 return {

 tag:"div",

 className:"info1",

 children:[{

 tag:"table",

 children:[

 {tag:"tr", children:[{tag:"td", html:"<b>Id</b>"}, {tag:"td", html:this.id}]},

 {tag:"tr", children:[{tag:"td", html:"<b>Label</b>"}, {tag:"td", html:this.label}]},

 {tag:"tr", children:[{tag:"td", html:"<b>Propriété 1</b>"}, {tag:"td", html:this.prop1}]},

 {tag:"tr", children:[{tag:"td", html:"<b>Propriété 2</b>"}, {tag:"td", html:this.prop2}]},

 {tag:"tr", children:[{tag:"td", html:"<b>Propriété 3</b>"}, {tag:"td", html:this.prop3}]},

 {tag:"tr", children:[{tag:"td", html:"<b>Propriété 4</b>"}, {tag:"td", html:this.prop4}]}

 ]

 }]

 };

 }

 }

 });

{

tag:"dashboard",

label:"Mon organisation",

labelsProperty:"myLabel",

childrenDisplayIds:[

 {

 id:"test1",

 expand:"expand",

 display:"obj",

 x:10,

 y:40,

 }

 ],

records:[

 {

 tag:"info1",

 id:"test1",

 myLabel:"Info 1",

 label:"Item 1",

 prop1:"aaa1",

 prop2:"bbb1",

 prop3:"ccc1",

 prop4:"ddd1"

 }

 ]

}

### Auto (valeur par défaut)

Le type de widget sera trouvé automatiquement :

* Si un objet correspondant au tag de l’enregistrement existe alors celui-ci sera utilisé pour afficher un ***form***.
* Si aucun objet n’est trouvé alors cela dépendra de l’existence ou non d’une propriété *records* qui correspond à un tableau, on affichera alors une ***grid*** sinon un ***properties***

## Modes d’ouverture d’un widget

On utilise la propriété ***expand*** :

### panel

Le widget apparaîtra déjà ouvert :



### popup (ou expand)

Le widget s’ouvre dans un cadre quand on clique dessus :



### tab

Similaire à *popup* mais peut s’ouvrir dans le rectangle englobant de notre choix.

Dans l’exemple ci-dessous on indique que le widget s’ouvre aux coordonnées (100, 200) et ait les dimensions 200 x 80 :



On utilise pour cela la propriété ***rect*** :



### wnd

Le widget s’ouvre sous la forme d’une fenêtre quand on clique dessus :



### dlgbox

Similaire à *wnd* mais avec des boutons *OK* et *Annuler* en bas :



### none

Le widget ne peut pas s’ouvrir :

