

On a : le gestionnaire de table(classe API) et les gestionnaire de cellule (classes : Gcell, GCellText, GCellSselect et GCellSelectFree)

La classe Api : c'est la classe qui gère la table ;

Propriétés

selector	Est égal à '#'+idTable
Data []	Contient les données sous forme tableau d'Object, chaque Object a plusieurs propriété, les propriétés correspondent au des colonnes
Columns []	Contient l'Object Gcell de chaque colonne
hideColumns []	Est un tableau de noms des colonnes cachées

Méthodes

getColumn (name index)	Récupérer le GCell à partir du nom du colonne ou de l'index
updateInput(i,name,value)	Met à jour la cellule à l'index « i » et à la colonne « name »
getTr(i)	Récupérer l'élément DOM de la ligne « i » en ajax
getTd(i,name)	Récupérer l'élément DOM de la cellule « i » « name » en ajax
getInput(i,name)	Récupérer l'élément DOM del'input de la lcellule « i » name » en ajax
getInputValue(name,i)	Recuperer la valeur de la cellule « i » « name »
addEmptyRow	Cette ajoute une nouvelle ligne

GCell : c'est la classe mère de la classe GCellText, GCellSelect

Propriétés

name	Le nom de la colonne
options	Contient les options de la cellule
isVisible	Définit la visibilité de la colonne correspondante
defaultValue	Contient la valeur par défaut lors du

Méthodes

getCellValue(i)	Récupérer la valeur de la cellule à l'index « i »
setCellValue(i,value)	Met à jour la valeur de la cellule à l'index « i »
Inputs()	Récupérer tous les ssséléments DOM Input en jq
update()	Mais a jour la collone après un modification de donnés direct
addEventInputBefore(action , function)	Ajouter un évènement avant le passage à l'editor
addEventInput(action , function)	Ajouter un évènement après le passage à l'editor
getDataCell(i)	Récupère les données(<> valeur) d'une cellule

