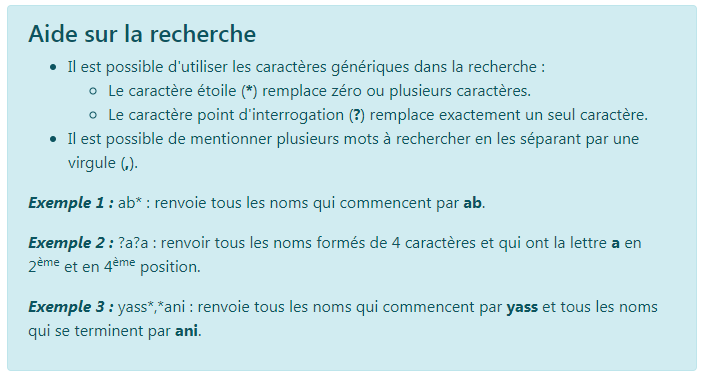
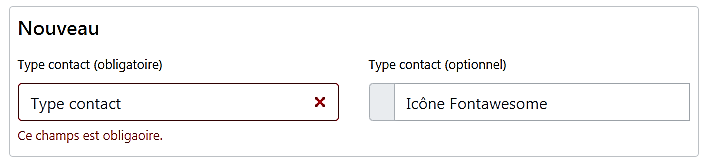
## 1. Components

### 1.1. aide-recherche



### 1.2. edit-filiale (need attention)

### 1.3. edit-type-contact



#### 1.3.1. Inputs

**@Input() typeContactCollection: TypeContactCollection;**

List of all type contacts in order to check for duplicates.

**@Input() typeContact = new TypeContact();**

The typeContact being edited by this component.

**@Input() insertMode = true;**

If insertMode is false than the component is in edit mode.

#### 1.3.2. Outputs

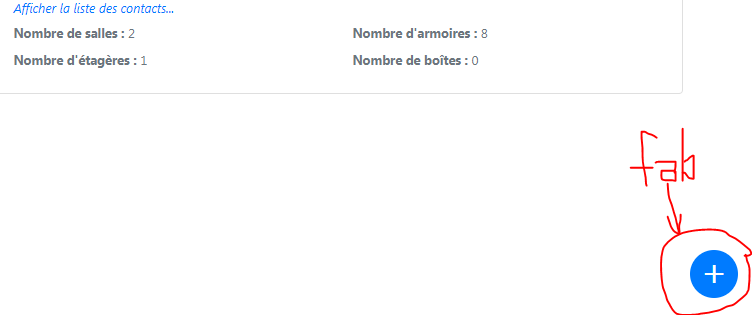
**@Output() submitData = new EventEmitter<TypeContact>();**

Submits the type contact that's being edited.

**@Output() cancelEditData = new EventEmitter<string>();**

Notifies the parent component that the user cancelled the edit operation.

### 1.4. fab



#### 1.4.1 Inputs (need attention)

**@Input() routerLink = [''];**

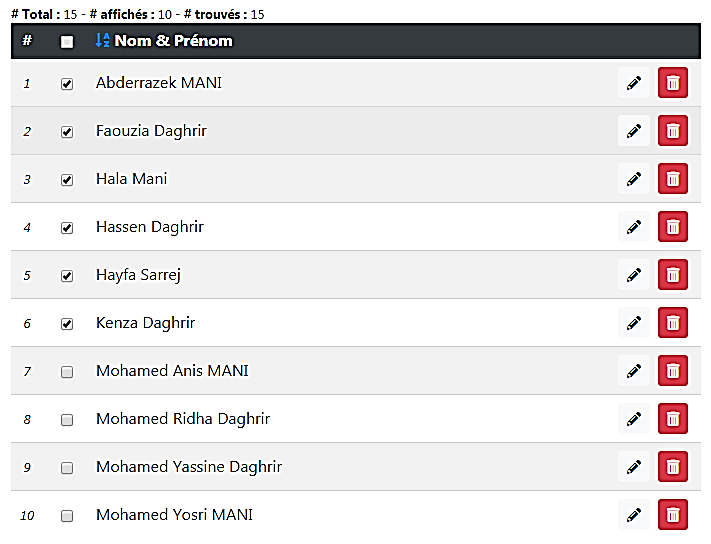
Not used yet.

#### 1.4.2 Outputs

**@Output() clicked: EventEmitter<any> = new EventEmitter();**

Notifies the parent component that the floating action button has been clicked.

### 1.5. table-responsables

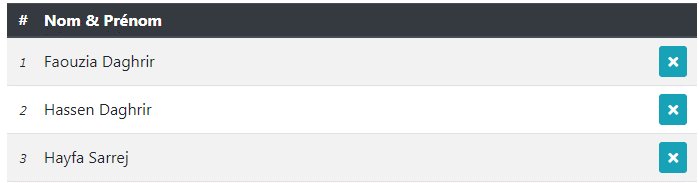


Operations column

edit

delete

Select/Deselect all items



remove

Select/Deselect one item

#### 1.5.1. Inputs

**@Input() contacts: Contact[] = [];**

An array of all contacts

**@Input() canSelect = true;**

Whether to display selection checkboxes or not.

**@Input() hasOperations = true;**

Whether to display the operation column or not.

**@Input() canEdit = true;**

Whether to display/hide the edit button.

**@Input() canDelete = true;**

Whether to display/hide the delete button.

**@Input() canRemove = false;**

Whether to display/hide the remove button, this button is meant to delete one contact from the selection list.

**@Input() canReorder = true;**

Determines if the user can order the table results.

**@Input() orderDir = 'ASC';**

Determines the order of items in the table of contacts.

#### 1.5.2. Outputs

**@Output() allItemsSelection = new EventEmitter<boolean>();**

Fired when the user clicks the table header checkbox to select or deselect all items.

**@Output() itemSelection = new EventEmitter<Contact>();**

Fired when the user selects one contact.

**@Output() orderChanged = new EventEmitter<string>();**

Fired when the user clicks the order link to change the ordering of contacts.

**@Output() editContactClicked = new EventEmitter<Contact>();**

Fired when the user clicks the edit button.

**@Output() deleteContactClicked = new EventEmitter<Contact>();**

Fired when the user clicks the delete button.

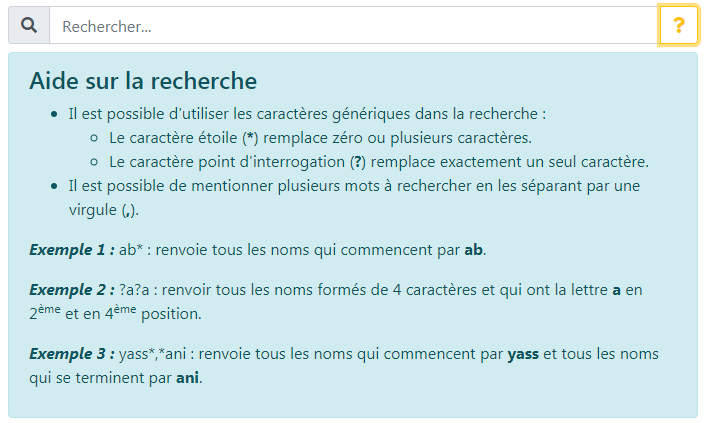
**@Output() removeContactClicked = new EventEmitter<Contact>();**

Fired when the user clicks the remove button.

### 1.6. search-field



Compact form



#### 1.6.1. Inputs

**@Input() searchPattern = '';**

Initialize the search pattern.

**@Input() helpDisplayed = false;**

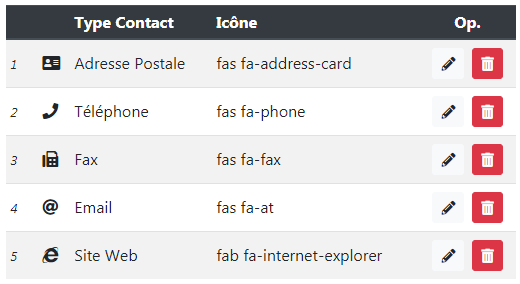
Tells if the help message should be displayed or hidden first time the component is rendered.

#### 1.6.2. Outputs

**@Output() searchPattenModified = new EventEmitter<string>();**

Fired when the user changes the search pattern.

### 1.7. table-types-contacts



#### 1.7.1. Inputs

**@Input() contactsTypes = new TypeContactCollection();**

The array of contact types objects.

**@Input() canEdit = true;**

Determines whether to show/hide the edit buttons.

**@Input() canDelete = true;**

Determines whether to show/hide the delete buttons.

**@Input() canReorder = true;**

Determines whether to show/hide the reorder button.

**@Input() orderedItemsIds = [];**

An array of the ids (numTypeContact) that was ordered.

**@Input() notOrderedItemsIds = [];**

An array of the ids (numTypeContact) that are not ordered yet.

#### 1.7.2. Outputs

**@Output() editButtonClicked = new EventEmitter<TypeContact>();**

Fired when the edit button is clicked.

**@Output() deleteButtonClicked = new EventEmitter<TypeContact>();**

Fired when the delete button is clicked.

**@Output() reorderButtonClicked = new EventEmitter<TypeContact>();**

Fired when the reorder button is clicked.

## 2. Services

## 3. Directives

### 3.1. FocusInputDirective

Use the directive **appFocusInput** to focus an input when a component becomes visible.

<input appFocusInput …>