Desarrollar un Componente reutilizable para reemplazar la lista de Appointments

**Agregar un UI Component al widget**

Acceder a la carpeta **statics/collection-training/src**

**Instalar el generador y generar un Nuevo UI Component**

npm install -g @backbase/generator-ui-ng

Correr:

bb-generate ui-ng

Responder las preguntas al generador de UI Component:

**Name** **==** **>** ui**-**training**-**appointments**-**list**-**ng

**Description** **-** Training Appointment List

**Version** **-** **---** Accept **default** value **---**

**Nota:** La **Description** es requerida y recuerda llevar la nomenclatura de creación del componente: ui-**name-project-name-widget-type[list,dropdown]**-ng.

Abrir el archivo **src/ui-training-appointments-list-ng/scripts/component.js** y renombrar los enlaces.

Enlaces actuales:

bindings**:** {

config**:** '<',

onSelect**:** '&',

}

Renombrar **config** a **items** para una mejor semántica:

bindings**:** {

items**:** '<',

onSelect**:** '&',

}

Reemplazar el **template** lo que se encuentra en la línea: <pre>{{ $ctrl.config | json }}</pre>

**template**: `

<pre>{{ $ctrl.config | json }}</pre>

`,

Por lo siguiente

<ul class**=**"list-group">

<li class**=**"list-group-item" ng-repeat**=**"item in $ctrl.items" ng-click**=**"$ctrl.onSelect({item: item})" ng-class**=**"{ 'urgent': item.urgent}">

<div class**=**"row row-eq-height">

<div class**=**"col-xs-5 col-sm-4 col-md-3 col-lg-2 right-border">

<div class**=**"text-center col-xs-3 col-sm-4 col-md-4"><span class**=**"fa fa-calendar fa-2x"></span></div>

<div class**=**"col-xs-9 col-sm-8 col-md-8 col-lg-8">

<div class**=**"text-left">

{{item.date | date:'dd/MM/yyyy'}}

</div>

<div class**=**"text-left">

{{item.date | date:'HH:mm'}}

</div>

</div>

</div>

<div class**=**"col-xs-7 col-sm-8 col-md-9 col-lg-10 overflow-hidden-on-xs">

<h5 ng-bind**=**"item.subject"></h5>

<div>

<div class**=**"col-xs-12 col-sm-3">

<h5>Guests:</h5>

</div>

<div class**=**"col-xs-12 col-sm-9">

<ul>

<li ng-repeat**=**"guest in item.guests">

<h6>

<span ng-bind**=**"guest.personalData.fullname"></span> <a href**=**"mailto:{{guest.personalData.email}}" class**=**"hidden-xs">&lt; {{guest.personalData.email}} &gt;</a>

</h6>

</li>

</ul>

</div>

</div>

</div>

</div>

</li>

</ul>

Cambiar el **template** de la forma anterior se puede realizar de otra forma más cómoda.

Agregar después de la primera línea, lo siguiente:

import template from '../template/template.ng.html';

Agregar una nueva carpeta en raiz del componente, denominada: **template** y dentro de esta agregar el archivo **template.ng.html** con el contenido html que anteriormente se agregó.

Quedando el script de la siguiente forma.

**import** controller **from** './controller';

**import** template **from** '../template/template.ng.html';

**const** component = {

controller,

bindings: {

items: '<',

onSelect: '&',

},

template,

};

**export default** component;

**Usando la extensión de UI Component de List Appointments.**

Abrir el archivo **statics/collection-training/src/ext-training-appointments-ng/templates/template.ng.html**

Reemplazar todo el contenido de **ul.list-group** por lo siguiente:

<ui-training-appointments-list-ng items**=**"$ctrl.appointments" on-select**=**"$ctrl.onSelectAppointments(item)"></ui-training-appointments-list-ng>

Abrirel archivo **statics/collection-training/src/ext-training-appointments-ng/scripts/index.js**

Buscar:

**export** **const** dependencyKeys **=** [

ngAriaModuleKey,

];

Reemplazarlo por lo siguiente:

**import** uiTrainingAppointmentsListKey from 'ui-training-appointments-list-ng';

**export** **const** dependencyKeys **=** [

ngAriaModuleKey,

uiTrainingAppointmentsListKey,

];

**Mejorar la apariencia del componente**

Los siguientes pasos ya se realizaron cuando creamos un **theme** desde cero para que nuestros widgets tengan los styles de Bootstrap.

1. Acceder a la carpeta **statics/collection-training/src/theme-training-default/styles/extensions**
2. Crear un archive y nombrarlo con **\_ext-training-appointments-ng.scss**
3. **Agregar** el siguiente codigo:

.**ext-training-appointments-ng** .**right-border** {

**border-right**: 1**px** **solid** **grey**;

}

.**ext-training-appointments-ng** .**urgent** {

**background-color**: #fff470;

}

Abrir el archivo **statics/collection-training/src/theme-training-default/styles/\_extensions.scss**

Agregar el siguiente codigo:

@**import** "extensions/ext-training-appointments-ng.scss";

Import to CXP Manager usando **bb-package** y **bb-import**

**npm run package-ng**

**npm run import-ng**

**Nota:** la funcionalidad puedes verla desde local en el Desarrollo de widgets: <http://localhost:8888/> únicamente acceder al widget que tiene el componente List.



**Implementar la funcionalidad de un Filtro con un Componente Dropdown**

Para implementar el filtro, necesitamos un Componente de interfaz de usuario desplegable para seleccionar un cliente y / o asesor para filtrar la lista.

**Generar Dropdown UI Component**

Acceder a la carpeta **statics/collection-training/src**

Correr.

bb-generate ui-ng

Responder las preguntas del geneardor de UI component:

**Name** **==** **>** ui**-**training**-**dropdown**-**ng

**Description** **-** Training Dropdown

**Version**  **-** **---** Accept **default** value **---**

Abrir el archivo **src/ui-training-dropdown-ng/scripts/component.js**

Agregar después de la primera línea, lo siguiente:

import template from '../template/template.ng.html';

Enlaces actuales:

bindings**:** {

config**:** '<',

onSelect**:** '&',

}

Reemplazar los enlaces por lo siguiente:

bindings**:** {

items**:** '<',

itemType**:** '<',

onSelect**:** '&',

}

El código del script debe quedar de la siguiente forma:

**import** controller **from** './controller';

**import** template **from** '../template/template.ng.html';

**const** component = {

controller,

bindings: {

items: '<',

itemType: '<',

onSelect: '&',

},

template,

};

**export default** component;

Agregar una nueva carpeta en raiz del componente, denominada: **template** y dentro de esta agregar el archivo **template.ng.html**

Al archivo **template.ng.html** agregar el siguiente código:

<div class**=**"dropdown" uib-dropdown>

<div>{{$ctrl.itemType}}:</div>

<button class**=**"btn btn-default dropdown-toggle" type**=**"button" id**=**"dropdownMenu1" data-toggle**=**"dropdown" uib-dropdown-toggle>

<span ng-if**=**"$ctrl.selected">{{$ctrl.selected.personalData.fullname}} </span> <span ng-if**=**"!$ctrl.selected">Select one {{$ctrl.itemType}} </span><span class**=**"caret"></span>

</button>

<ul class**=**"dropdown-menu" role**=**"menu" aria-labelledby**=**"dropdownMenu1" uib-dropdown-menu>

<li ng-click**=**"$ctrl.select(null)">

<a role**=**"menuitem" tabindex**=**"-1" href**=**"#" >Deselect</a>

</li>

<li ng-repeat**=**"item in $ctrl.items" ng-click**=**"$ctrl.select(item)">

<a role**=**"menuitem" tabindex**=**"-1" href**=**"#" ng-bind**=**"item.personalData.fullname"></a>

</li>

</ul>

</div>

Abrir el **controller.js** y agregar el siguiente código al método: **$onInit**

**const** $ctrl **=** **this**;

$ctrl.selected **=** **null**;

$ctrl.select **=** **function** (item) {

$ctrl.selected **=** item;

$ctrl.onSelect({item**:** item});

};

Abrir el **index.js** e importar las librerías de vendor para el dropdown:

**import** uibDropdownKey from 'vendor-bb-uib-dropdown';

...

**export** **default** angular

.module('ui-training-dropdown-ng', [uibDropdownKey])

**Usando el componente de Dropdown**

Abrir el archive **statics/collection-training/src/ext-training-appointments-ng/templates/template.ng.html**

**Reemplazar** el contenido por el siguiente código:

<section class**=**"ext-training-appointments-ng" ng-controller**=**"AppointmentsController as $ctrl">

<div class**=**"container-fluid">

<div class**=**"row">

*<!-- DROPDOWNS -->*

<ui-training-dropdown-ng

class**=**"col-sm-6" items**=**"$ctrl.customers"

item-type**=**"$ctrl.customersHeader"

on-select**=**"$ctrl.onSelectCustomer(item)"></ui-training-dropdown-ng>

<ui-training-dropdown-ng

class**=**"col-sm-6" items**=**"$ctrl.advisors"

item-type**=**"$ctrl.advisorsHeader"

on-select**=**"$ctrl.onSelectAdvisor(item)"></ui-training-dropdown-ng>

</div>

<div class**=**"row">

*<!-- LIST COMPONENT -->*

<ui-training-appointments-list-ng items**=**"$ctrl.appointments"

on-select**=**"$ctrl.onSelectAppointments(item)"></ui-training-appointments-list-ng>

<h4 ng-if**=**"$ctrl.appointments.length === 0">No appointments available.</h4>

</div>

</div>

</section>

**Usando la lógica del filtro.**

Extraer el archivo **utils.zip** que se encuentra en **BB-Training/exercise** en **statics/collection-training/src/widget-training-appointments-ng/scripts**

Abrir el archivo **statics/collection-training/src/widget-training-appointments-ng/scripts/controller.js**

Al comienzo del archivo agregar el siguiente código.

**import** filter from './utils/filter';

Buscar:

**const** $ctrl **=** **this**;

Debajo de la línea **agregar** lo siguiente:

$ctrl.filters **=** {

advisor**:** **null**,

customer**:** **null**

};

$ctrl.advisorSelected **=** **null**;

$ctrl.customerSelected **=** **null**;

$ctrl.advisorsHeader **=** 'Advisor';

$ctrl.customersHeader **=** 'Customer';

$ctrl.customers **=** [];

$ctrl.advisors **=** [];

$ctrl.appointments **=** [];

$ctrl.filteredAppointments **=** [];

Buscar:

$ctrl.appointments **=** appointments;

Debajo de la línea **agregar**:

$ctrl.filteredAppointments **=** appointments.concat();

Buscar:

items

**Reemplazar** por:

appointments

**Implementando el evento select para el método del Dropdown**

Buscar:

**const** $onInit **=** () => {

**Después** del método **agregar**:

**const** onSelectCustomer **=** (customer) => {

**const** $ctrl **=** **this**;

$ctrl.filteredAppointments **=** filter({

filtered**:** $ctrl.filteredAppointments,

original**:** $ctrl.appointments,

filters**:** $ctrl.filters,

item**:** customer,

name**:** 'customer',

comparator**:** 'advisor'

});

$ctrl.customerSelected **=** customer;

};

**const** onSelectAdvisor **=** (advisor) => {

**const** $ctrl **=** **this**;

$ctrl.filteredAppointments **=** filter({

filtered**:** $ctrl.filteredAppointments,

original**:** $ctrl.appointments,

filters**:** $ctrl.filters,

item**:** advisor,

name**:** 'advisor',

comparator**:** 'customer'

});

$ctrl.advisorSelected **=** advisor;

};

Buscar:

**$onInit,**

Debajo de la línea **agregar**:

onSelectCustomer,

onSelectAdvisor,

Abrir el archivo **statics/collection-training/src/ext-training-appointments-ng/templates/template.ng.html**

Buscar:

$ctrl.appointments

**Reemplazar** todos los que hagan match por:

$ctrl.filteredAppointments

**Inyectar el componente Dropdown en la vista del widget**

Abrir **statics/collection-training/src/ext-training-appointments-ng/scripts/index.js**

Buscar:

**import** uiTrainingAppointmentsListKey from 'ui-training-appointments-list-ng';

Debajo de la línea **agregar**:

**import** uiTrainingDropdownKey from 'ui-training-dropdown-ng';

Buscar:

uiTrainingAppointmentsListKey,

Debajo de la línea **agregar**:

uiTrainingDropdownKey,

Importar al CXP Manager usando **bb-package** y **bb-import**

**npm run package-ng**

**npm run import-ng**