

Para aplicar los filtros vamos a usar dos librerías `sap/ui/model/Filter` y `sap/ui/model/FilterOperator` que deben ser declaradas en el controlador de la vista:

Bootcamp-SAP-FIORI > freestyle > webapp > controller > `JS` Home.controller.js > `sap.ui.define()` callback > `onPress`

You, 4 minutes ago | 2 authors (You and others)

```

1 sap.ui.define([
2   "sap/ui/core/mvc/Controller",
3   "com/bootcamp/sapui5/freestyle/utils/HomeHelper",
4   "sap/ui/model/Filter", ←
5   "sap/ui/model/FilterOperator" ←
6 ], (Controller, HomeHelper, Filter, FilterOperator) => {
7   "use strict";
8

```

## Filtro a nivel de aplicación

Con la tabla construida anteriormente vamos a agregar el control “Input” que nos permitirá ingresar el valor a filtrar

Bootcamp-SAP-FIORI > freestyle > webapp > view > `Home.view.xml`

```

5  >
6  <Page
7    id="page"
8    title="{i18n>title}"
9  >
10 <VBox id="vTest">
11   <Text
12     id="textPrueba"
13     text="{i18n>textPruebai18n}"
14   />
15   <Toolbar id="toolbarID">
16     <Input id="inputID" change="onChange"/> ←
17     <Button
18       id="btnBuscar"
19       text="{i18n>textBuscar}"
20       press="onPress"
21     />
22   </Toolbar>
23   You, 15 minutes ago • Uncommitted changes
24 </VBox>

```

A este control le implementamos el evento `change` implementado en la función `onChange`. Este evento se dispara al ingresar un valor y presionar enter:

Bootcamp-SAP-FIORI > freestyle > webapp > controller > **JS** Home.controller.js > ...

```

6  ], (Controller, HomeHelper, Filter, FilterOperator) => {
38   onChange: function(oEvent){
39     let oFilter = [];
40     let oSource = oEvent.getSource();
41     let oTable = this.getView().byId("idProductsTable")
42     let oBinding = oTable.getBinding("items");
43
44     if(oSource.getValue()){
45       oFilter = new Filter("ProductID", FilterOperator.EQ, oSource.getValue());
46     }
47     oBinding.filter(oFilter);
48   });
49

```

Explicación:

1. oFilter, definición del objeto que contendrá el filtro
2. Al ser un evento del control podemos obtener referencia del mismo con el getSource()
3. Se obtiene la referencia de la tabla
4. Se obtiene los items de la tabla
5. Luego con un getValue() podemos obtener el valor ingresado y crear el filtro usando las librerías `sap/ui/model/Filter` y `sap/ui/model/FilterOperator`
6. Por último se aplica el filtro a los items de la tabla

Product	Unit Price	Units In Stock	Quantity Per Unit
Chef Anton's Gumbo Mix	21.35 USD	0	36 boxes

## Filtro a en el backend

Para esto vamos a usar el mismo input pero este tomará el valor ingresado una vez presionen el botón de buscar (deshabilitamos el evento change)

```
Bootcamp-SAP-FIORI > freestyle > webapp > controller > JS Home.controller.js > ...
6  ], (Controller, HomeHelper, Filter, FilterOperator) => {
38     onChange: function(oEvent){
39         // let oFilter = [];
40         // let oSource = oEvent.getSource();
41         // let oTable = this.getView().byId("idProductsTable")
42         // let oBinding = oTable.getBinding("items");
43
44         // if(oSource.getValue()){
45             // oFilter = new Filter("ProductID", FilterOperator.EQ, oSource.getValue());
46         // }
47         // oBinding.filter(oFilter);
48     }
49 );
```

En el evento press vamos a agregar lo siguiente

```
Bootcamp-SAP-FIORI > freestyle > webapp > controller > JS Home.controller.js > sap.ui.define() callback > onItemPress > [oDatos]
6  ], (Controller, HomeHelper, Filter, FilterOperator) => {
14     onPress: async function(){
15
16         let oFilter = [];
17         let sValue = this.byId("inputID").getValue();
18
19         if(sValue){
20             oFilter = new Filter("ProductID", FilterOperator.EQ, sValue)
21         }
22
23         let oDatos = await HomeHelper.getDataProducts([oFilter]);
24         await HomeHelper.setProductModel(this, oDatos[0].results);
25     },
26 }
```

Explicación:

1. oFilter, definición del objeto que contendrá el filtro
2. Se obtiene la referencia del control input
3. Si hay valor con un `getValue()` podemos obtener el valor ingresado y crear el filtro usando las librerías `sap/ui/model/Filter` y `sap/ui/model/FilterOperator`
4. Se modifica la función `getDataProducts` para pasar el filtro como parámetro

```
Bootcamp-SAP-FIORI > freestyle > webapp > utils > JS HomeHelper.js > sap.ui.define() callback > setProductModel
4  ], function (HomeService, JSONModel) {
5
6
7
8
9
10    },
11
12    getDataProducts: async function(oFilters) {
13        //let oFilters = [];
14        return HomeService.readProducts(this._oNorthwindModel, oFilters);
15    },
16 }
```

Freestyle SAPUI5			
Control Text de Prueba con i18n			
Product	Unit Price	Units In Stock	Quantity Per Unit
8 Northwoods Cranberry Sauce	40.00 USD	6	12 - 12 oz jars