Creo 2 Fragment

1) ProductSimple – ProductExtended  
  
ProductSimple:  
<core:FragmentDefinition

    xmlns="sap.m"

    xmlns:core="sap.ui.core">

    <StandardListItem

        id="productSimple"

        icon="sap-icon://warning"

        title="{products>ProductName} ({products>QuantityPerUnit})"

        info="{i18n>Discontinued}"

        type="Active"

        infoState="Error"

        press=".onItemSelected">

        </StandardListItem>

</core:FragmentDefinition>  
  
ProductExtended:  
<core:FragmentDefinition

    xmlns="sap.m"

    xmlns:core="sap.ui.core">

    <ObjectListItem

        id="productExtended"

        title="{products>ProductName} ({products>QuantityPerUnit})"

        number="{

            parts: [

                {path: 'products>UnitPrice'},

                {path: 'customModel>/currencyCode'}

            ],

            type: 'sap.ui.model.type.Currency',

            formatOptions: { showMeasure: false }

        }"

        type="Active"

        numberUnit="{customModel>/currencyCode}"

        press=".onItemSelected">

        </ObjectListItem>

</core:FragmentDefinition>

2)Chiamiamo I Fragment nella view:  
  
    xmlns:core="sap.ui.core" – dipendenza per fragment

  <List id="list3"

            headerText="{i18nproductListTitle}"

            items="{

                path: 'products>/Products',

                factory: '.productListFactory'

            }">

               <dependents>

                <core:Fragment

                    fragmentName="project1.view.fragments.ProductSimple"

                    type="XML"/>

                <core:Fragment

                    fragmentName="project1.view.fragments.ProductExtended"

                    type="XML"/>

            </dependents>

        </List>

2)Scrivo la funzione factory nel controller:  
  
AGGIORNO DIPENDENZE:  
  "sap/m/ObjectAttribute"

],

    /\*\*

     \* @param {typeof sap.ui.core.mvc.Controller} Controller

     \*/

    function (Controller, Text, mobileLibrary, Locale, LocaleData, Currency ,  ObjectAttribute) {

  productListFactory: function(sId, oContext) {

                var oUIControl;

                // Decide based on the data which dependent to clone

                if (oContext.getProperty("UnitsInStock") === 0 && oContext.getProperty("Discontinued")) {

                    // The item is discontinued, so use a StandardListItem

                    oUIControl = this.byId("productSimple").clone(sId);

                } else {

                    // The item is available, so we will create an ObjectListItem

                    oUIControl = this.byId("productExtended").clone(sId);

                    // the item is temporarily out of stock, so we will add a status

                    if (oContext.getProperty("UnitsInStock") < 1) {

                        oUIControl.addAttribute(new ObjectAttribute({

                            text : {

                                path: "i18n>outOfStock"

                            }

                        }))

                    }

                }

                return oUIControl;

            }