In the step 36, there is a small confusion:

Graphical user interface, text, application

Description automatically generated

It says **HelloDialog.controller.js** while it is **HelloPanel.controller.js**

Also, the **Fragment** module doesn’t exist anymore in the example, but the code still works;

I assume it is not loaded as a fragment anymore?! But there is **.loadFragment**,which calls the loadFragment API. So, is it a fragment still, or not? (to be continued…)

So, in the end, in the **HelloPanel.controller.js**, this is the code I’m using:

(I’ve added just the use of pDialog promise response, in onOpenDialog)

Graphical user interface, text, application

Description automatically generated

sap.ui.define([

    "sap/ui/core/mvc/Controller",

    "sap/m/MessageToast",

    "sap/ui/core/Fragment",

    "sap/ui/core/syncStyleClass"

 ], function (Controller, MessageToast, Fragment, syncStyleClass) {

    "use strict";

    return Controller.extend("sap.ui.demo.walkthrough.controller.HelloPanel", {

        onShowHello : function () {

                // read msg from i18n model

                var oBundle = this.getView().getModel("i18n").getResourceBundle();

                var sRecipient = this.getView().getModel().getProperty("/recipient/name");

                var sMsg = oBundle.getText("helloMsg", [sRecipient]);

                // show message

                MessageToast.show(sMsg);

        },

        onOpenDialog : function () {

            // create dialog lazily

            if (!this.pDialog) {

                this.pDialog = this.loadFragment({

                    name: "sap.ui.demo.walkthrough.view.HelloDialog"

                }).then(function (oDialog){

                    // forward compact/cozy style into dialog

                    syncStyleClass(this.getOwnerComponent().getContentDensityClass(), this.getView(), oDialog);

                    return oDialog;

                }.bind(this));

            }

            this.pDialog.then(function(oDialog) {

                oDialog.open();

            });

        },

        onCloseDialog : function () {

            // note: We don't need to chain to the pDialog promise, since this event-handler

            // is only called from within the loaded dialog itself.

            this.byId( "helloDialog" ).close();

        }

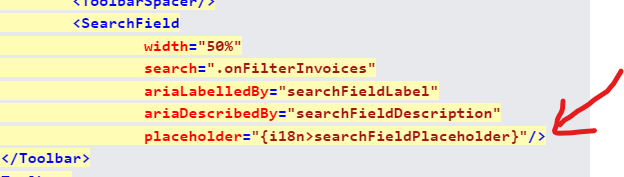
    });  });

**On step 37: Accessibility**

Inside webapp/i18n/i18n.properties there is one extra line needed – the one which describes the **placeholder** of Search input.

It is used in webapp/view/InvoiceList.view.xml, but never actually registered:

(nor in the available code on platform: <https://sapui5.hana.ondemand.com/#/entity/sap.m.tutorial.walkthrough/sample/sap.m.tutorial.walkthrough.37/code/webapp/i18n/i18n.properties> )



So we can just add it in the i18.properties:

#Search Control

searchFieldPlaceholder=Search