

## **Oracle Visual Builder Redwood Cookbook 2025-2026**

### **(Chapter 27).**

### **Workaround for customization when add layouts feature is not available in SCM Sales Orders Redwood pages.**

#### **Introduction/Goal**

Some Redwood pages allow, to include new layouts, which means new calculated information, new buttons to run processes, etc.

This feature provides great flexibility to achieve user's requirements.

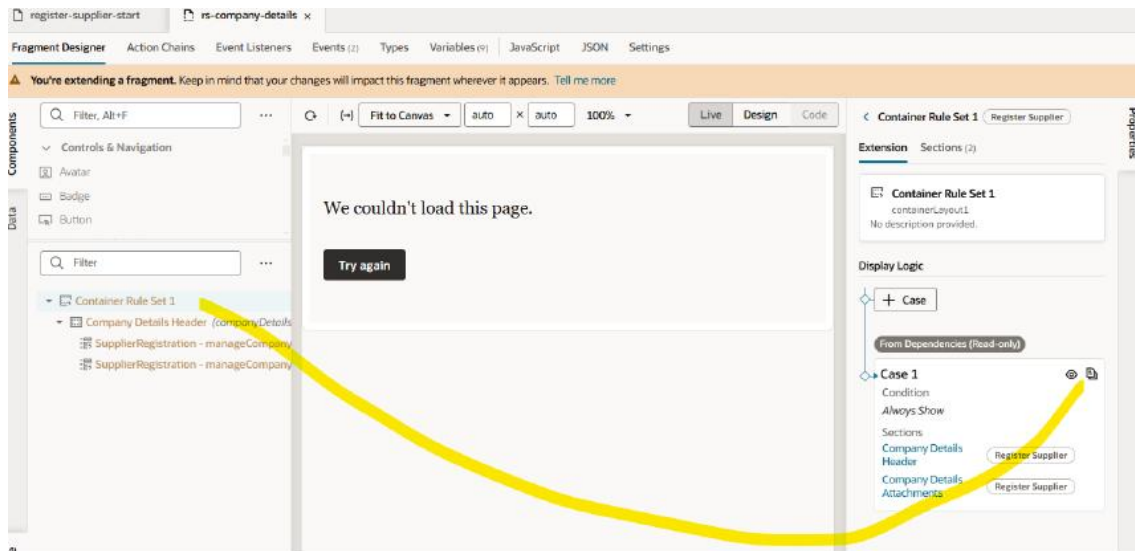
The following screenshot shows a dummy example, in the responsive Suppliers page, notice that we are able to add new fields, buttons, etc.:

The screenshot displays a Redwood page layout with the following components:

- A dropdown menu labeled "Supplier Type" with "Supplier" selected.
- A text area labeled "Justification".
- A section header "Attach tax, insurance, and other relevant documents".
- A dropdown menu labeled "Category" with "From Supplier" selected.
- A "Drag and Drop" zone with the text "Select or drop files here.".
- A text input field labeled "URL" and a button labeled "Add URL".
- A section header "demoSectionHead".
- Two buttons labeled "Button Demo1" and "Button Demo2".
- Two text input fields, both containing the text "Hello, the selected business unit id is:300000046987012".
- Three buttons at the bottom right: "Cancel", "Save", and "Continue".

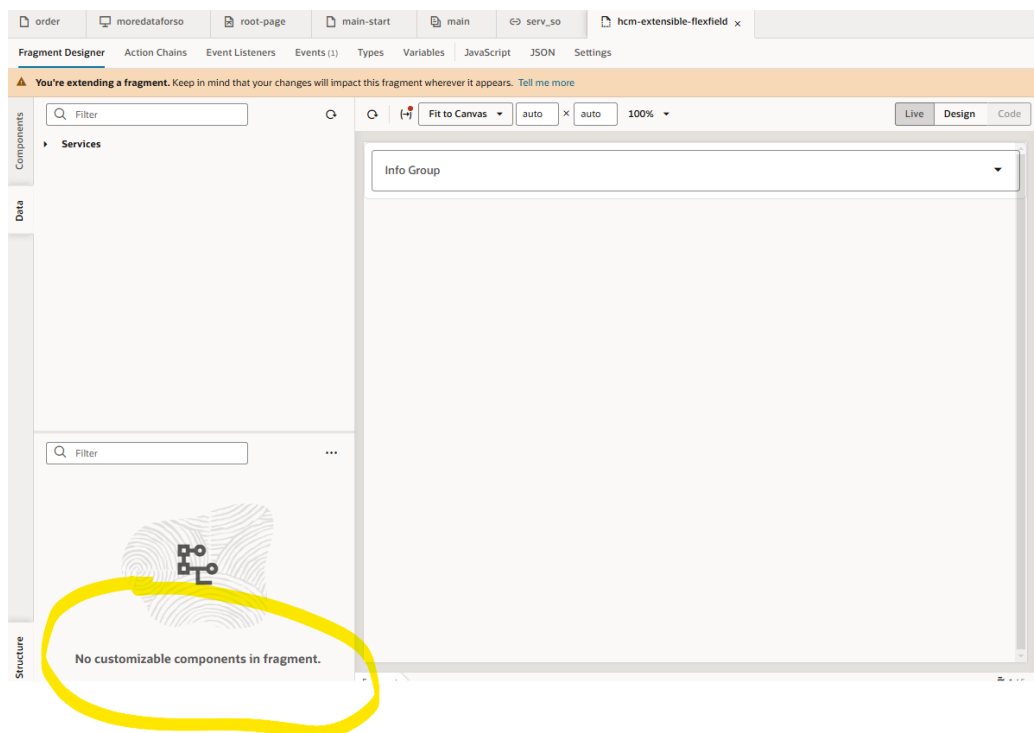
Yellow arrows point to the "Button Demo1" and "Button Demo2" buttons, highlighting the ability to add new UI elements.

A way to know if Redwood pages support new layouts is review if you are able to duplicate "Cases" as shown in next screenshot:



On the Sales Orders page (in SCM module) we have not found how to add new layouts, so on paper it is not feasible to add new fields/buttons/processes.

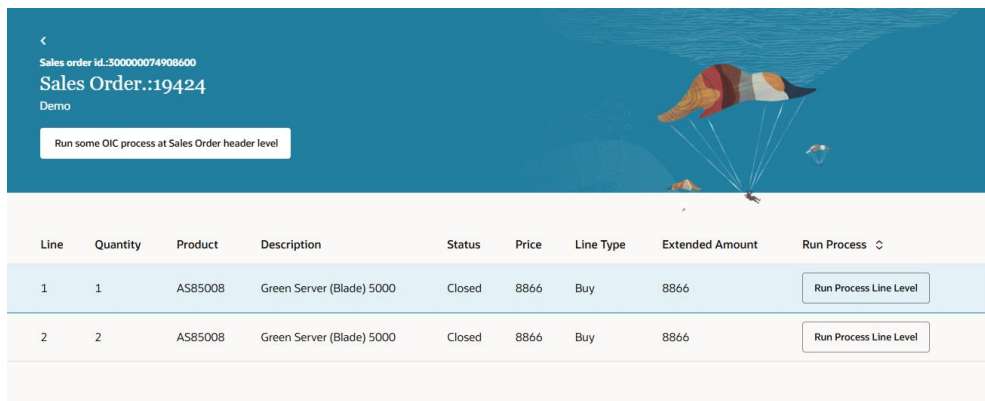
We have been reviewing the page with no success:



If we have a requirement to show additional information related to the Sales Order, what can we do?

**Visual Builder and JavaScript to the rescue!!!!**

We will provide the user with a page like this:



Sales order id:30000074908600  
Sales Order.:19424  
Demo

Run some OIC process at Sales Order header level

Line	Quantity	Product	Description	Status	Price	Line Type	Extended Amount	Run Process
1	1	AS85008	Green Server (Blade) 5000	Closed	8866	Buy	8866	Run Process Line Level
2	2	AS85008	Green Server (Blade) 5000	Closed	8866	Buy	8866	Run Process Line Level

Somewhere I have read, Visual Builder was planned as a “no-code” or “low-code” tool, but my opinion is that to get complex requirements done you need to code, and this takes time to learn.

Also to get some complex requirements done you will need JavaScript, if you do not use it at all there will be requirements that will not be possible to accomplish.

In Visual Builder you will not need to be a JavaScript expert, but you will need some basics and cut and paste/trial error approach (mine).

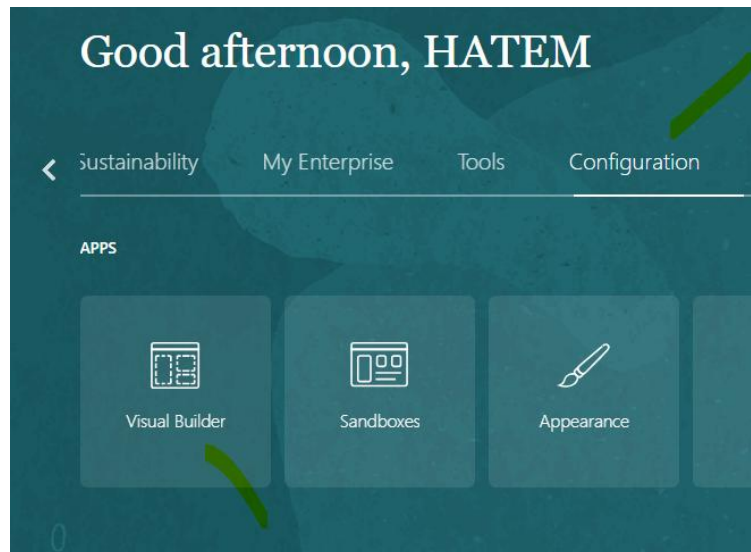
As much as you learn JavaScript will make your coding with Redwood Visual Builder easier.

**For learning purposes**, I will show a **workaround** to get the job done, it is not clean, is tricky, but works and could provide ideas to others to get a better approach, while Oracle developers working in Redwood provide a place to include new layouts in these pages.

## Preparatory steps

Access Oracle ERP with a user with roles to see sales orders and Visual Builder customization.

Navigate to Configuration-Visual Builder:



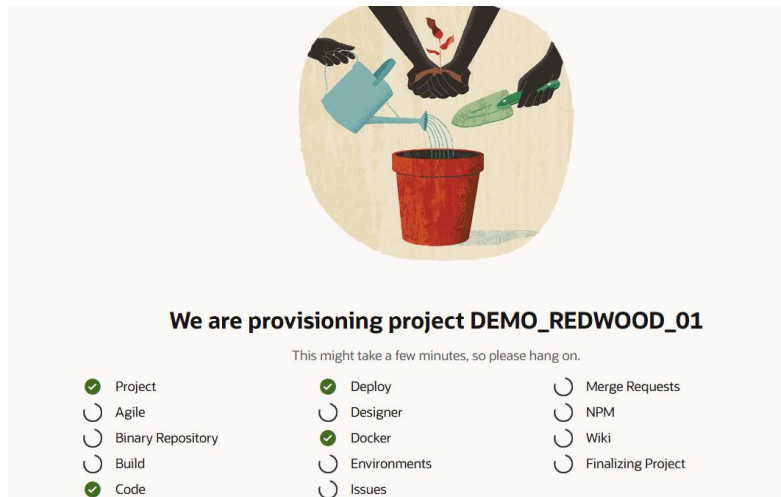
Create project.

A screenshot of the "New Project" dialog box. It has a title bar "New Project" with a close button. Below the title bar is a progress bar with four steps: 1. Details, 2. Template, 3. Properties, and 4. Team. The "Details" step is currently active. Below the progress bar, there's a section titled "Project Details" with two input fields: "Name" (containing "DEMO\_REDWOOD\_01") and "Description".

Press Next.

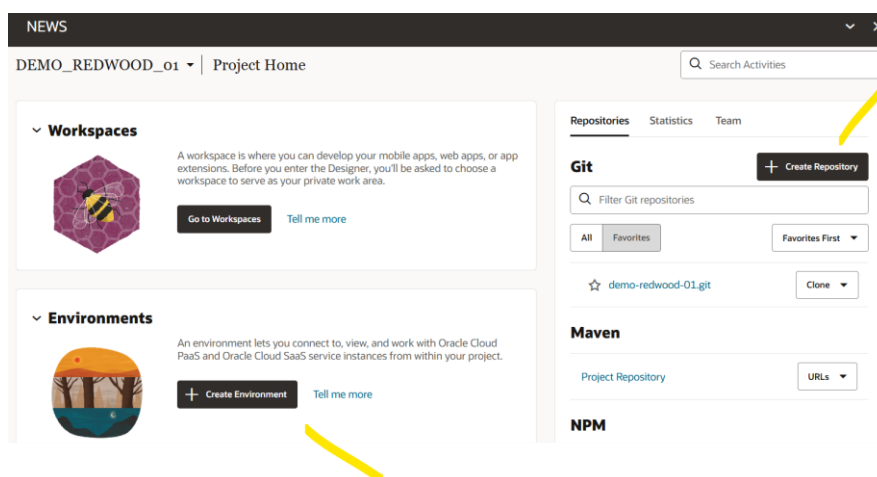
A screenshot of the "New Project" dialog box, now on the "Template" step. The progress bar shows step 2 is active. Below the progress bar, there's a section titled "Project Template" with a subtitle "Optionally create your project from a template to clone its settings, including any git repository content and build jobs." There are two options: "Empty Project" (with a trash icon) and "Initial Repository" (with a folder icon). The "Initial Repository" option is highlighted with a blue background.

Drink your coffee, it will take a little while to finish.



To import provided code (see below the 3 files for different phases) and I will show you the steps I have followed.

Create repository and environment:



To create repository:

New Repository

Name

repository\_01

Description

Initial content

☐ Empty repository

☒ Initialize repository with README file


☐ Import existing repository

Cancel

Create

And to create environment.

Workspaces



A workspace is where you can develop your mobile apps, web apps, or app extensions. Before you enter the Designer, you'll be asked to choose a

Environment Name


FA

Description

Cancel

Create

Environments

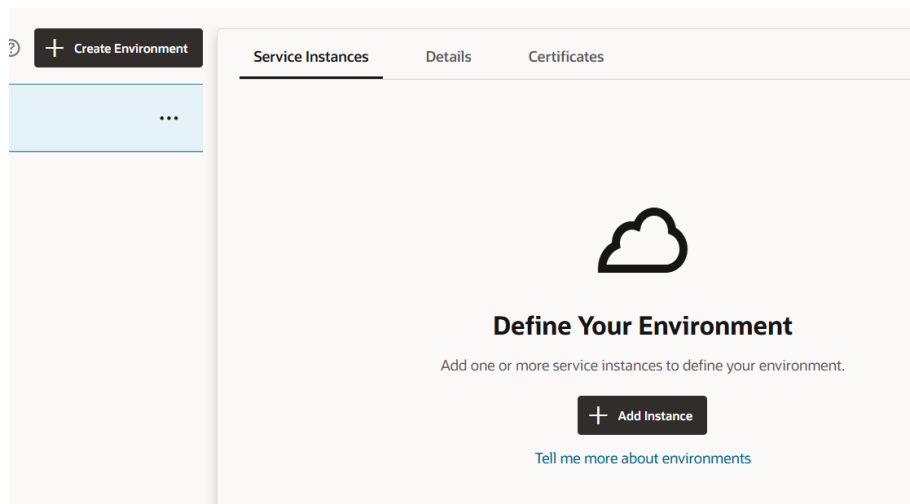


+ Create Environment

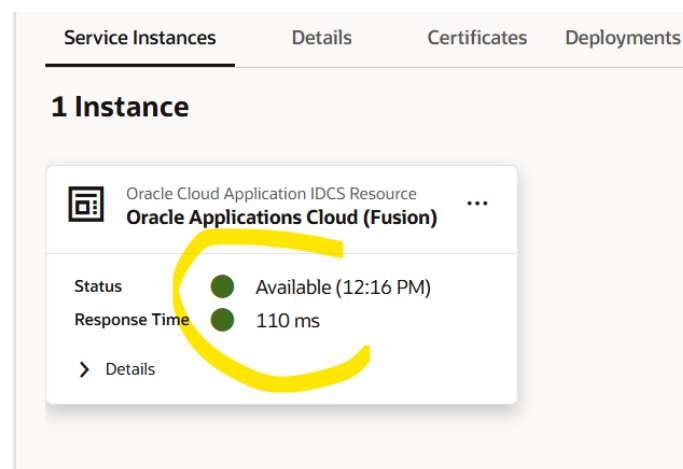
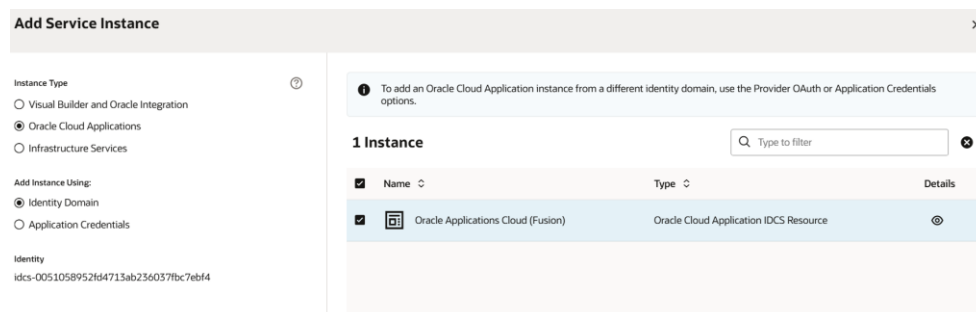
[Tell me more](#)

Click "Add Instance".

6 | Oracle Visual Builder Redwood Cookbook 2025



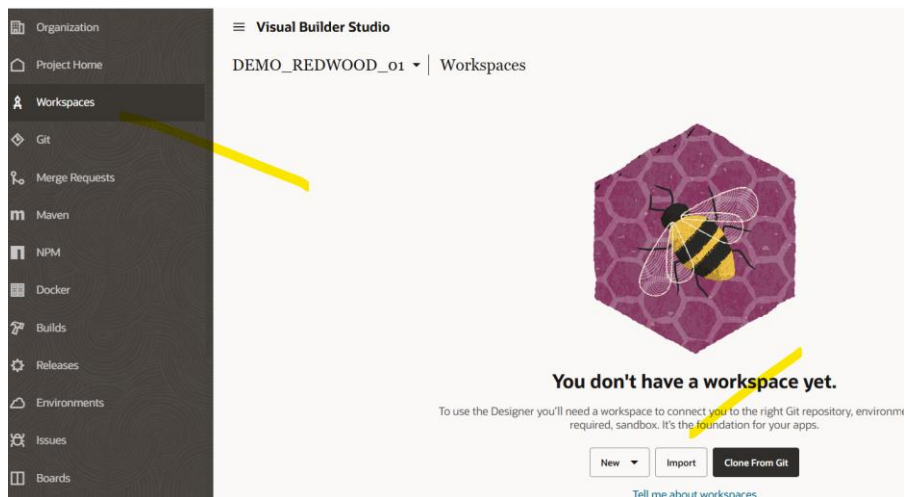
Select as shown:



Now import provided example files see below the differences.

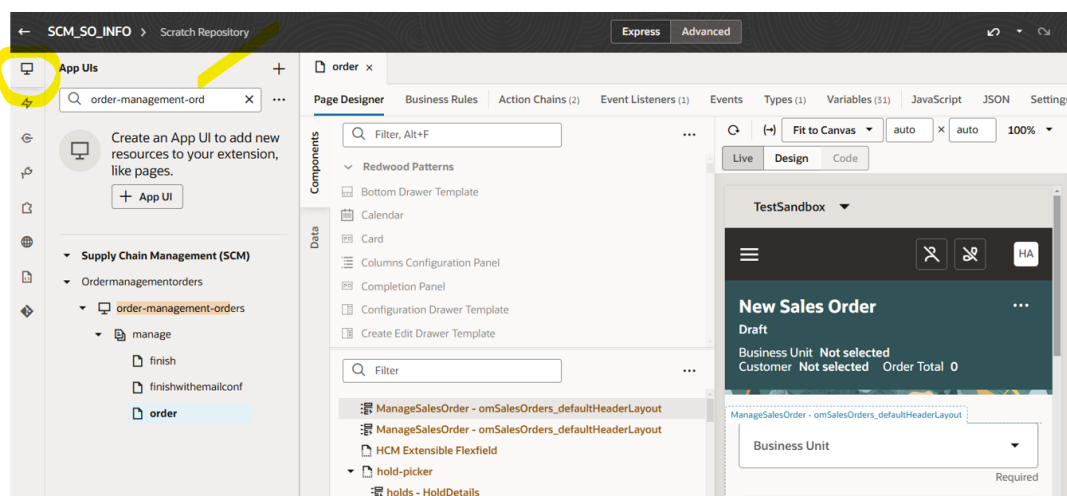
Download from: [https://github.com/juanjesusmontero/oracle\\_redwood\\_cookbook](https://github.com/juanjesusmontero/oracle_redwood_cookbook)

Navigate to Workspaces, Import.



Assign a name:

Optional: filter as shown to take a look at the customization features.



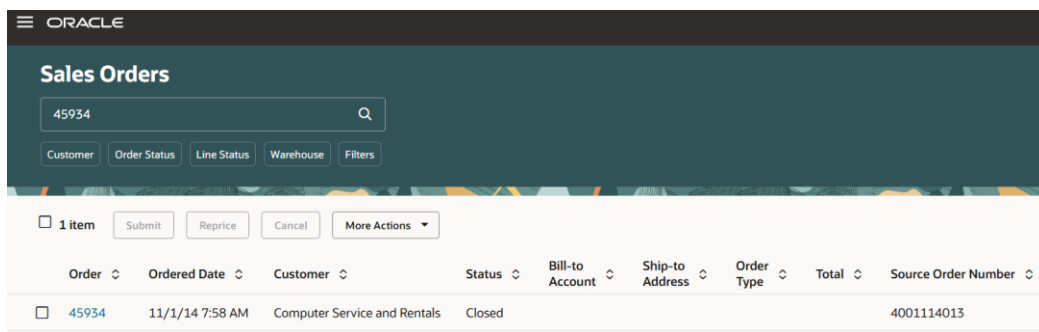
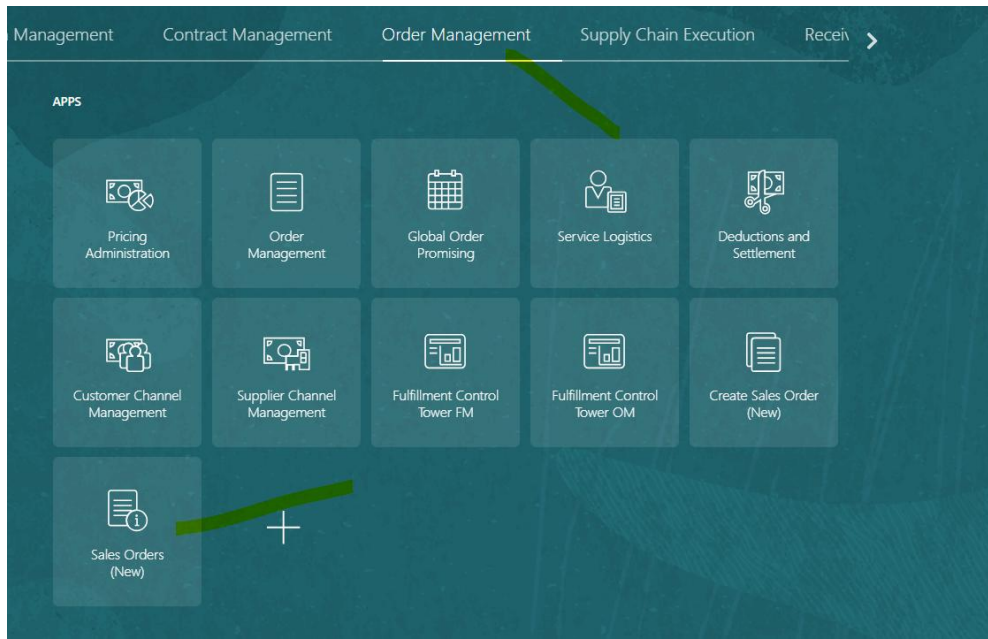
And now close the browser.



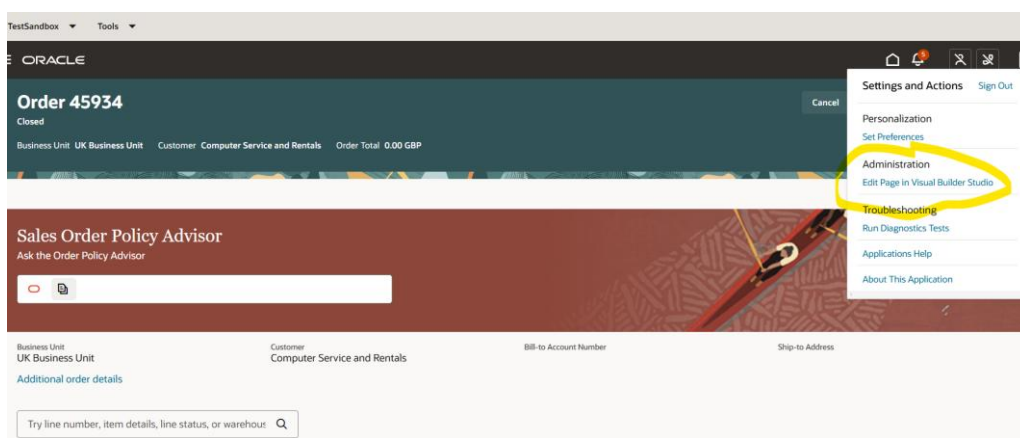
There is a reason for this, if we run the page in preview just now, we will not have a context sales order, and to have one is good for testing using preview features, in next section we will open a “Sales order” in ERP-CLOUD to get the context and see our changes.

## Navigation

Navigate to Order Management and in our case query 45934 order.



Open Visual Builder.



Select our project.

Select Project

×

Q

Filter Projects

+

Create

Your Projects

You're already a member of these projects, so you can choose one and get to work right away:

DEMO\_REDWOOD\_01

Recommended

No description

SI

SCM IMPL [scm\_impl]  
scmimpl@gmail.com

Project Owner

CRM Sample

No description

SI

SCM IMPL [scm\_impl]  
scmimpl@gmail.com

Project Owner

HI

demo\_jj

SCM IMPL [scm\_impl]

Cancel

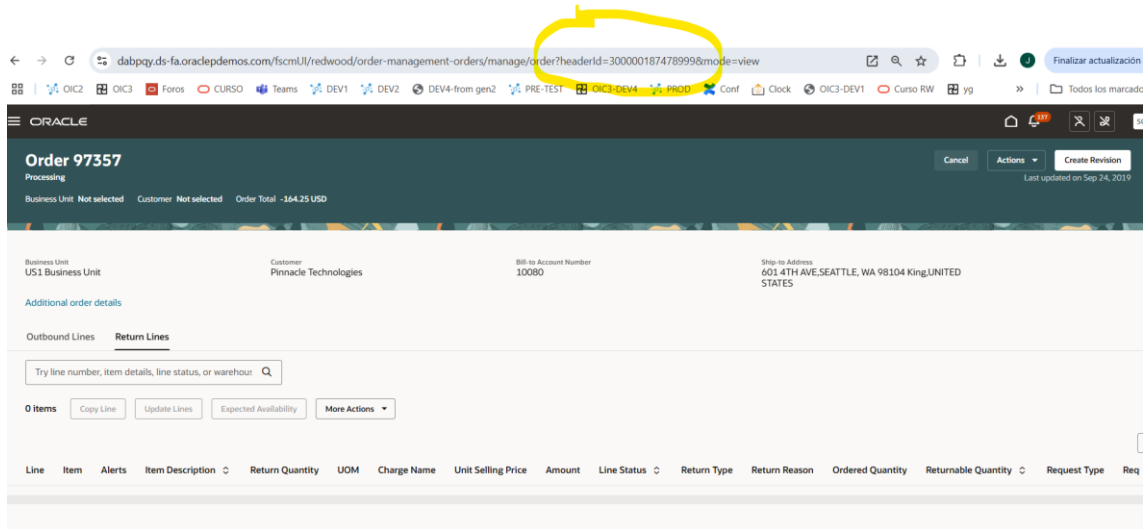
Select

## Summary of steps

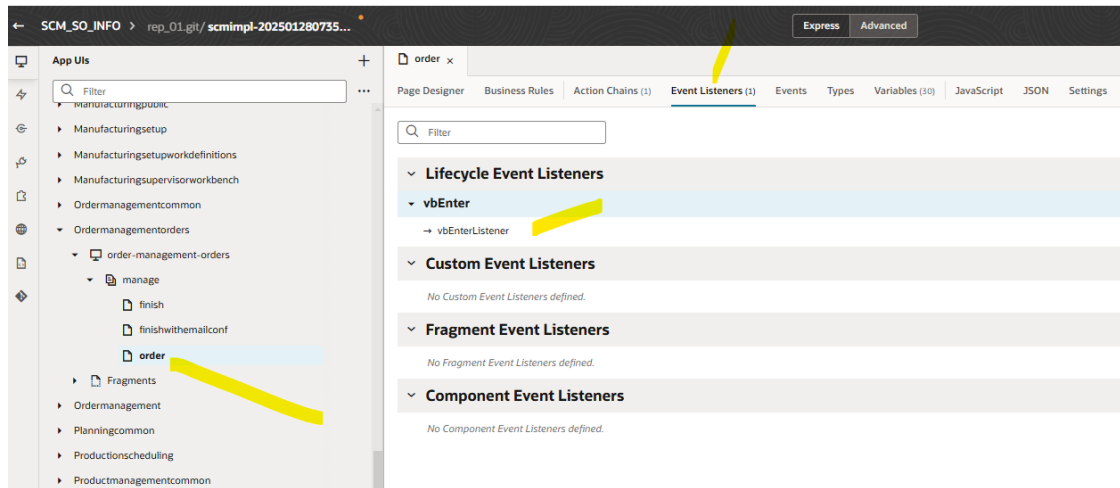
We will not explain steps sequentially but will focus on the artifacts we have used.

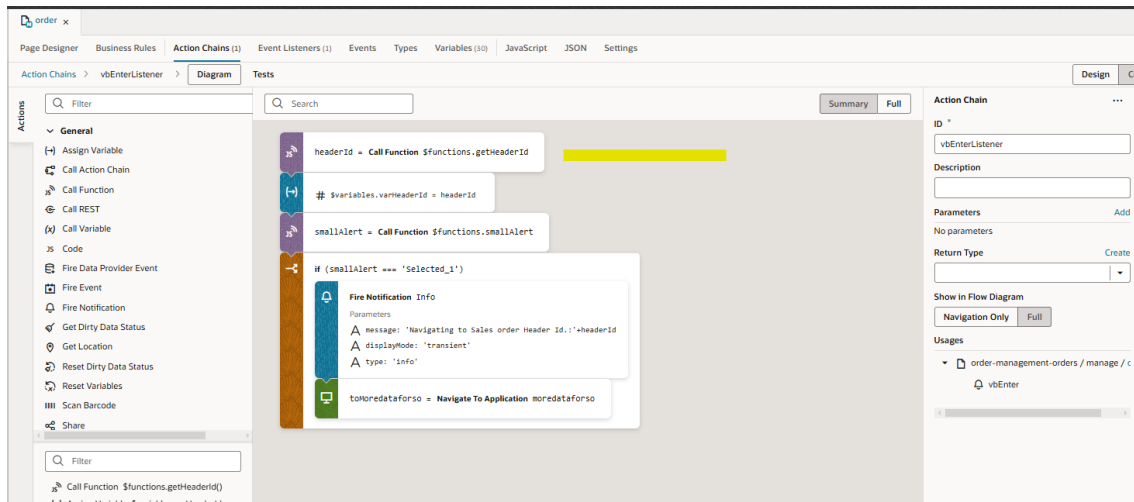
The starting point of the whole process is that we have found that the URL of the page contains the Header Id. of the Sales Order.

With JavaScript code we will find it and use it in the whole process, it is the entry door to do whatever we need.

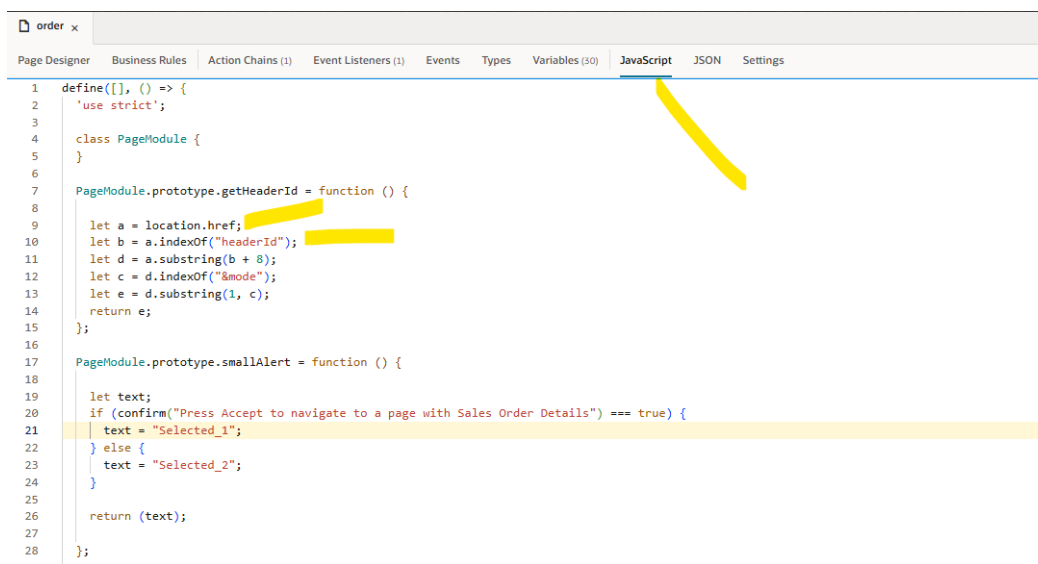


The magic is in “vbEnterListener” Event Listener and a bit JavaScript code.

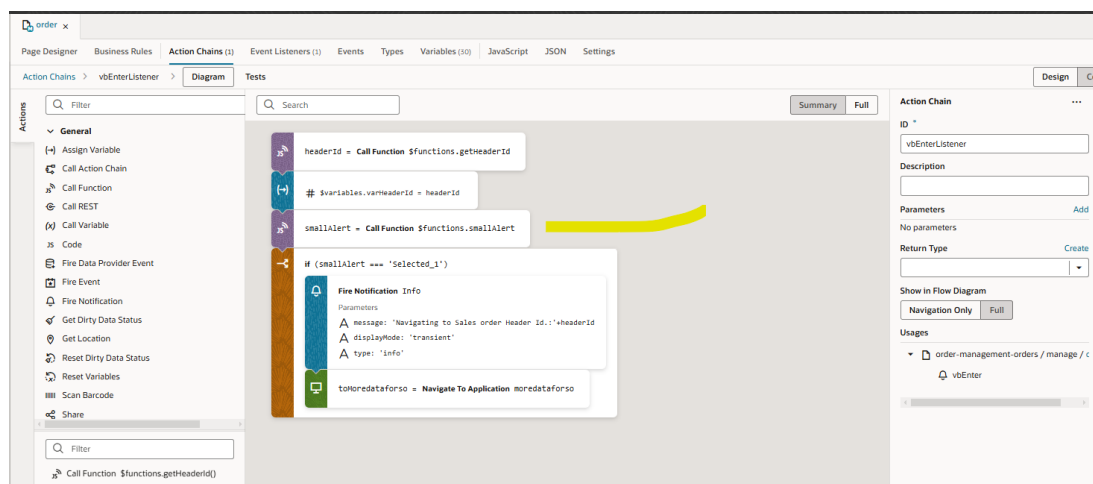




In this piece of JavaScript, we get the URL and find the “headerId”.



Here we ask the user If he/she wants to navigate to the page with additional features.



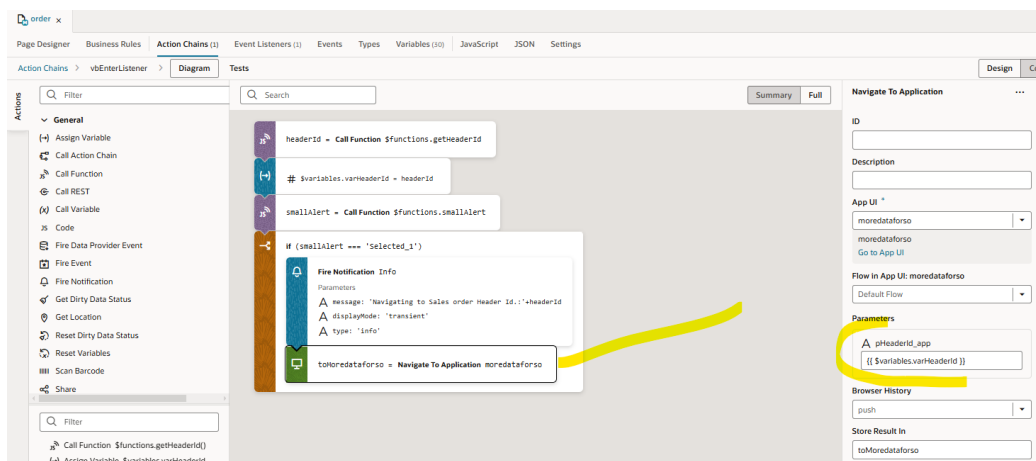
And this is the piece of JavaScript to show the question.

```

1  define([], () => {
2    'use strict';
3
4    class PageModule {
5    }
6
7    PageModule.prototype.getHeaderId = function () {
8
9      let a = location.href;
10     let b = a.indexOf("headerId");
11     let d = a.substring(b + 8);
12     let c = d.indexOf("&mode");
13     let e = d.substring(1, c);
14     return e;
15   };
16
17   PageModule.prototype.smallAlert = function () {
18
19     let text;
20     if (confirm("Press Accept to navigate to a page with Sales Order Details") === true) {
21       text = "Selected_1";
22     } else {
23       text = "Selected_2";
24     }
25
26     return (text);
27   };
28 }

```

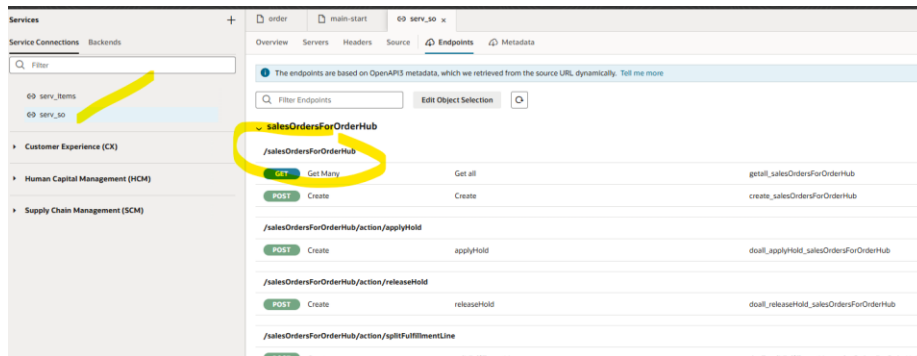
Here we navigate to the new page sending the parameter (HeaderId):



We have built from scratch a page using Welcome Page Template.

This is an extension that can be built using the free Visual Builder embedded in Fusion Applications.

You will need a connection for the REST API that manages the Sales order.



We find this is the correct REST searching in the reference documentation.

- ▶ Sales Order Action Requests
- ▶ Sales Orders for Order Hub
  - Apply hold
  - Create sales orders
  - Delete one sales order
  - Get all sales orders
  - Get one sales order
  - Release hold
  - Split fulfillment lines
  - Update one sales order
  - Update scheduling attributes
- ▶ Attachments
- ▶ Bill-to Customers
- ▶ Copy Orders
- ▶ Details
- ▶ Holds
- ▶ Order Lines
- ▶ Payments
- ▶ Sales Credits

## Sales Orders for Order Hub REST Endpoints

Sort by Task Path Method

### Order Management/Sales Orders for Order Hub

The Sales Orders for Order Hub resource gets, creates, and updates sales orders in Oracle Order Management Cloud. A sales order captures order header details, such as customer, items ordered, and pricing.

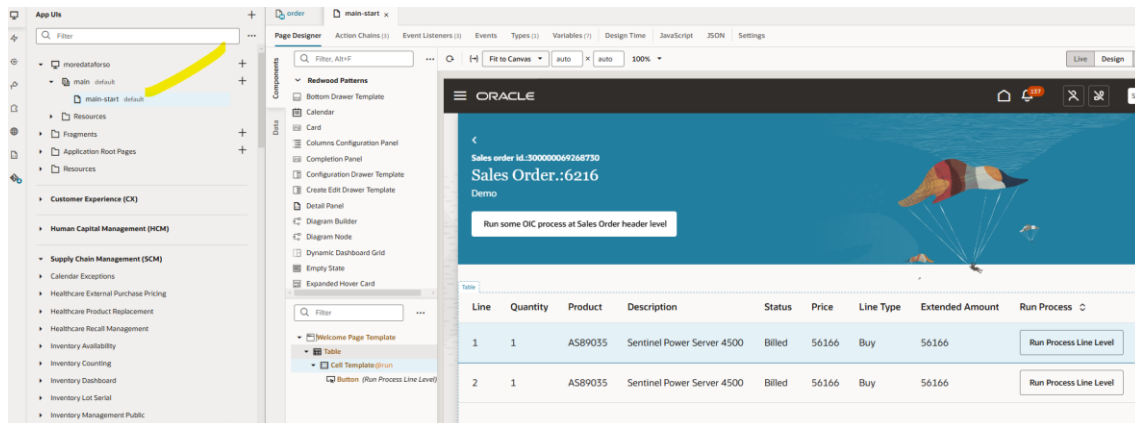
Here are the differences in REST API compared to the Order Import web service:

- To extend the processing logic to set the default values and to validate and transform the order during order creation and submission through REST API, create an order management extension. The Sales Orders for Order Hub resource doesn't support the pre-transformation, product transformation, and post-transformation rules.
- The Sales Orders for Order Hub resource also doesn't support automatic creation of customer, address, or contact information while creating an order. The resource also doesn't handle any cross-referencing of the primary and reference data.

Common error handling:

- 404 Not Found: Ensure that the correct parameter is used in the URL; for example, for linesUniqId, use the hash-generated key. Additionally, this error can occur if the user that is used while invoking the GET and Updating the order doesn't have business unit data security granted.
- 500 Internal Server Error: Ensure that the EFF deployment is completed successfully or there are no errors encountered while deploying the flexfield.
- 504 Gateway Timeout: The Order request may still be processed after the gateway time out; however, ensure that there are no validation failures that the order runs into while processing. You may need to run GET action to validate that the order is created.

[Apply hold](#)



Some properties of the root node.

**Welcome Page Template** ...

**General** Events (2) All

**ID**

oj-sp-wp1

**Page Title \***

[[ 'Sales Order.' + \$variables.varSoNumber ]]

**Overline**

[[ 'Sales order id.' + \$application.variables.pHeaderId\_app ]]

**Description**

Demo

**Background Color**

[[ \$page.variables.backgroundColor ]]

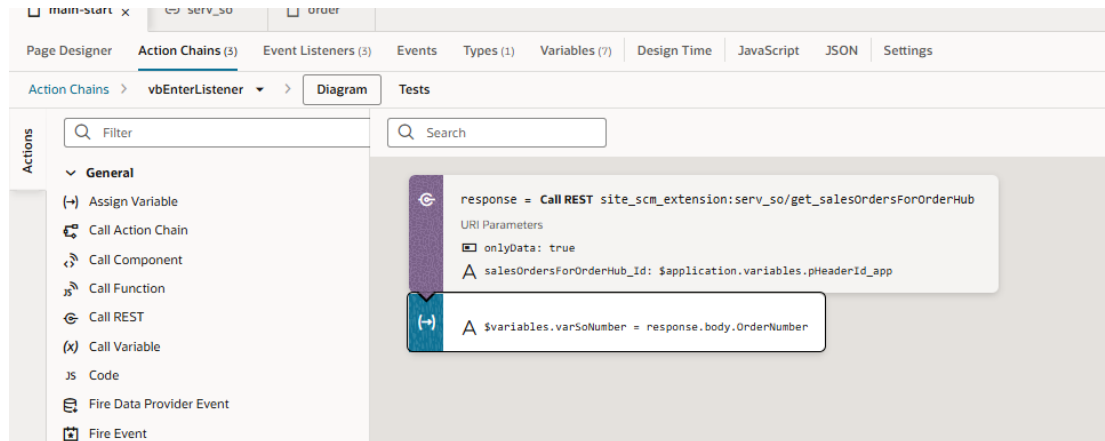
I will have a variable of type parameter at application level (after testing other levels for this variable do not work as expected):

The screenshot shows the 'Variables' tab in the Oracle Visual Builder configuration interface. The 'pHeaderId\_app' variable is highlighted with a yellow arrow. The 'Access for Application Extensions' section shows 'Read/Write' permissions, and the 'Input Parameter' section shows 'Required' checked.

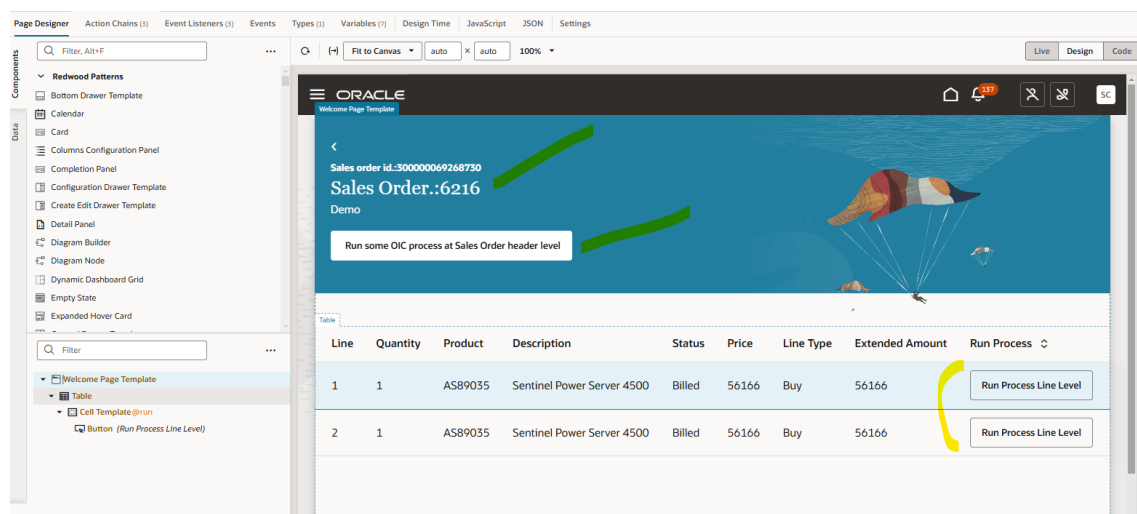
In the called page we will also have “vbEnterListener” Event Listener, to retrieve additional data.

The screenshot shows the 'Event Listeners' tab in the Oracle Visual Builder configuration interface. The 'vbEnter' event listener is highlighted with a yellow arrow, and the 'vbEnterListener' action is visible.

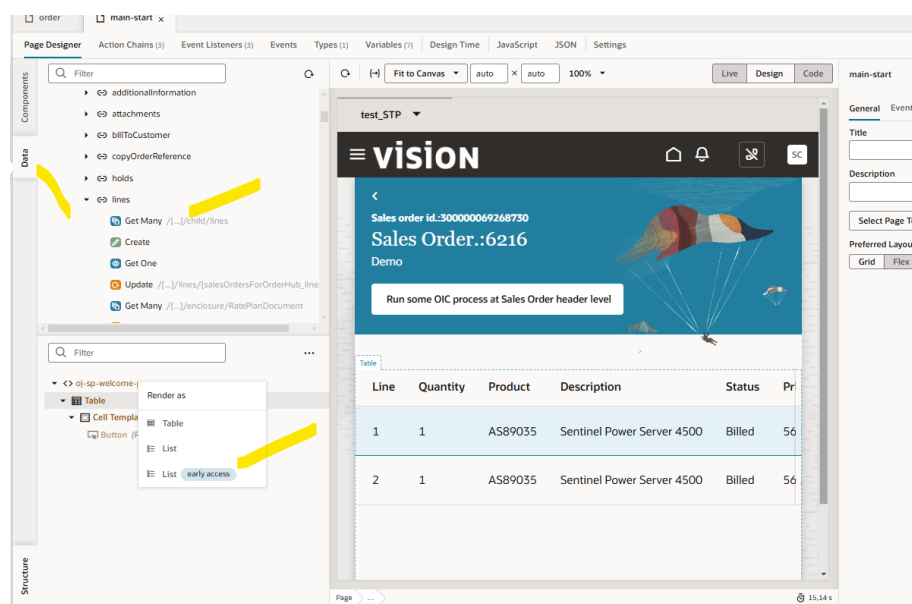




In the new page, we could show additional information at header or line level and run processes at header or line level.



We have added a table for Sales Order line using drag and drop from “Data”, Get Many using the Table option.



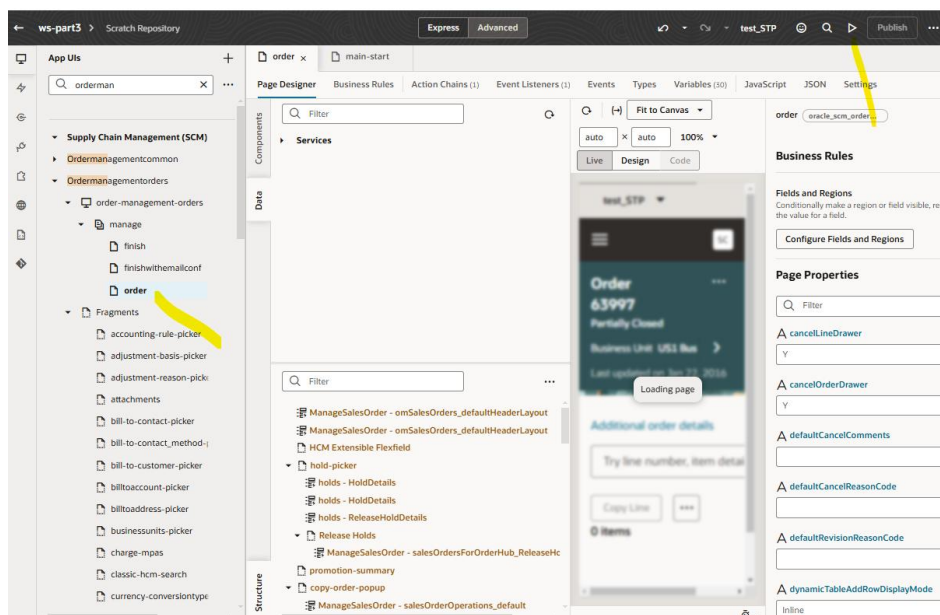
Feel free to review the code provided.

### Conclusion-Testing.

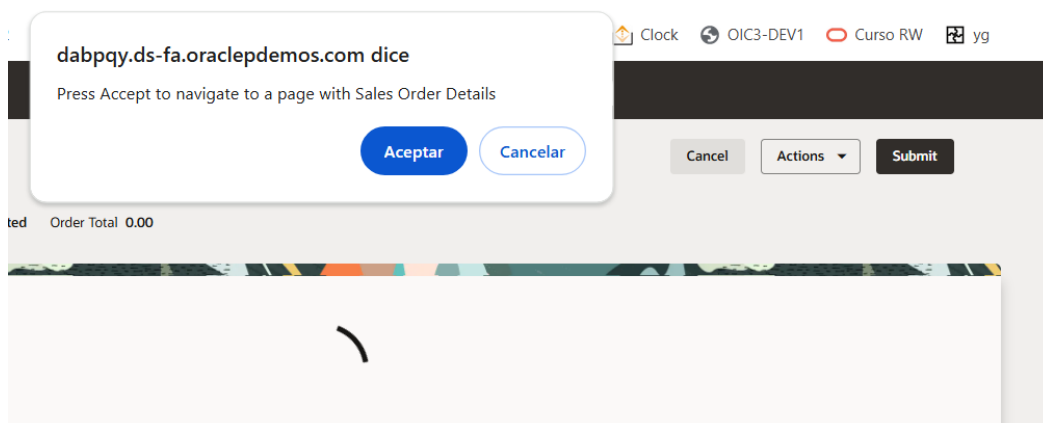
We have overcome the limitations the Sales Order page have (in the future this feature could be provided by Oracle).

We could now add new data and/or run processes for Sales Order and lines.

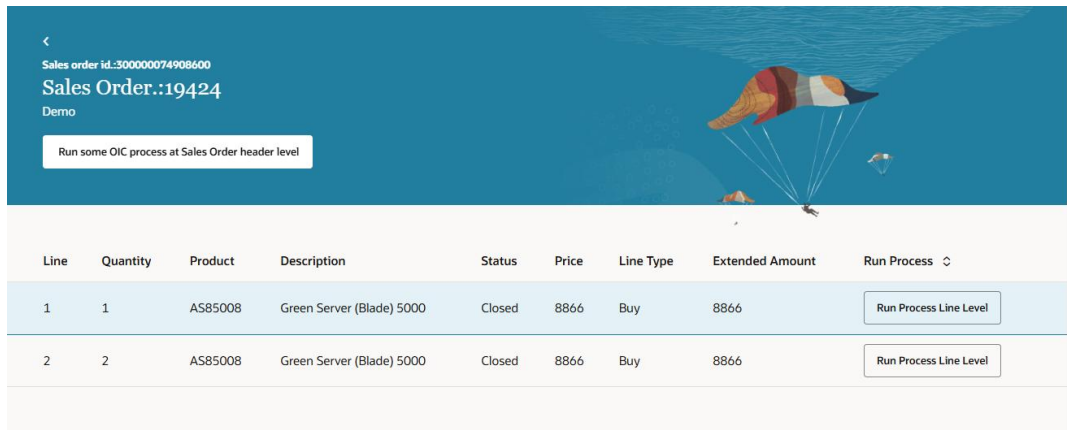
We could publish (see reference oracle documentation) our extension or for quick testing we can run in preview mode.



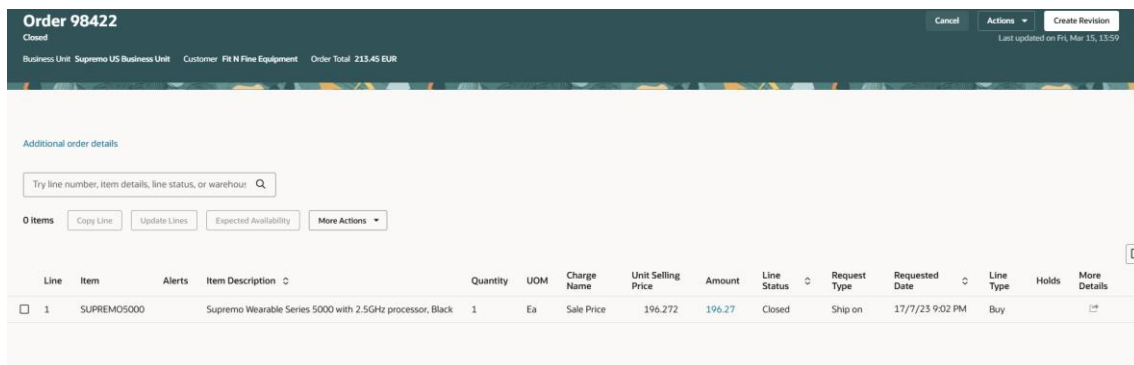
When we access a Sales Order, we would see a basic question run by our JavaScript code.



If user press “Accept” we would navigate to the new page.



If cancel the flow will go on and show the current page:



### **Files provided:**

The evolution of our testing is reflected in the following files you could import at your convenience.

Download from: [https://github.com/juanjesusmontero/oracle\\_redwood\\_cookbook](https://github.com/juanjesusmontero/oracle_redwood_cookbook)

### **Part 1**

Summary: we will show a message with Sales Order Id.

The code to import is in file: blog\_01\_part1\_fra\_demo-redwood-01\_19323\_1-1.0.zip

### **Part 2**

Summary: we will navigate to a new page built from scratch that will be able to show the Sales order number.

The code to import is in file: blog\_01\_part2\_fra\_demo-redwood-01\_19323\_1-1.0.zip

### **Part 3**

Summary: we will improve the new page to show how new data could be added and how process at header level or a line level could be accomplished.

The code to import is in file: blog\_01\_part3\_fra\_demo-redwood-01\_19323\_1-1.0.zip