

SCM-Inventory. Part II.

Visual Builder Redwood Custom pages to scan a EAN13 barcode.

Extension developed with VB included in Fusion (the free one).

Introduction/Goal

Requirements as follows:

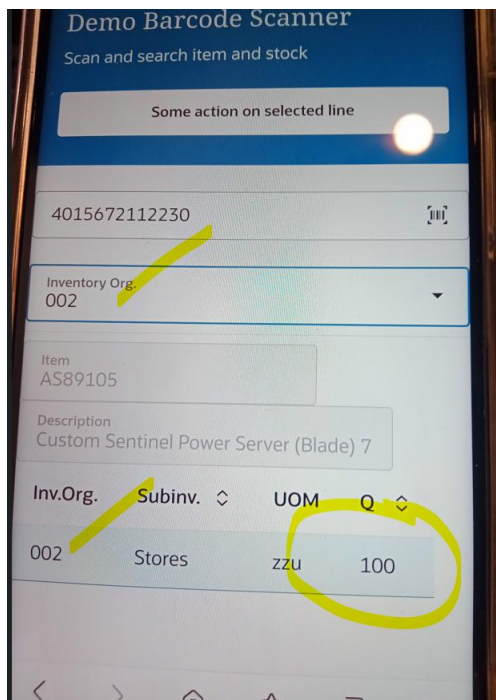
1. Scan an item EAN13 barcode. PART I
2. Show stock using REST. PART II (this one).

Details:

- a) Show item+description and
- b) User will be able to select Inventory Organization to limit the stocks shown

This document will be “Part 2” of this demonstration (Part 1 in document Redwood_CookBook 2026 SCM ITEM SCANNER.pdf).

In next picture we will see the result of this (Part II) exercise using a mobile (more pictures to show evidence in testing section).



Preparatory steps.

1. Roles.

You will need a user with proper roles to access Visual Builder in Fusion and to create Sandbox.

For publishing-deployment you should have the FND_ADMINISTER_SANDBOX_PRIV privilege in the target instance.

There are plenty of blogs with related information about the roles needed to work with Visual Builder, SaaS and Sandboxes.

Basic list of roles to consider in SaaS to get DEVELOPER_ADIMINSTRATOR ROLE in VB:

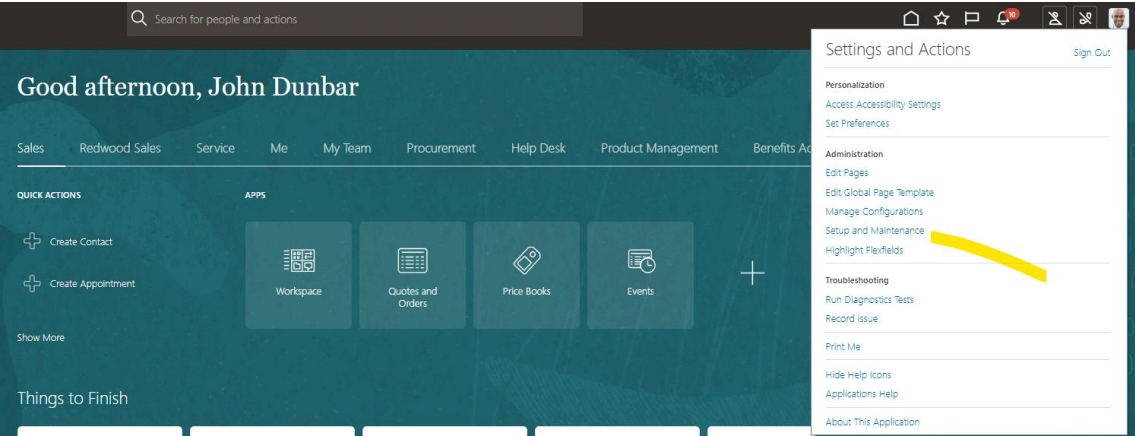
- Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB)
- Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB)
- Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)
- Synchronization Enabled Administrator Identity (ORA_SYNC_ENABLED_ADMINISTRATOR_ABSTRACT)

Basic list of roles to consider in SaaS to get DEVELOPER_USER ROLE in VB:

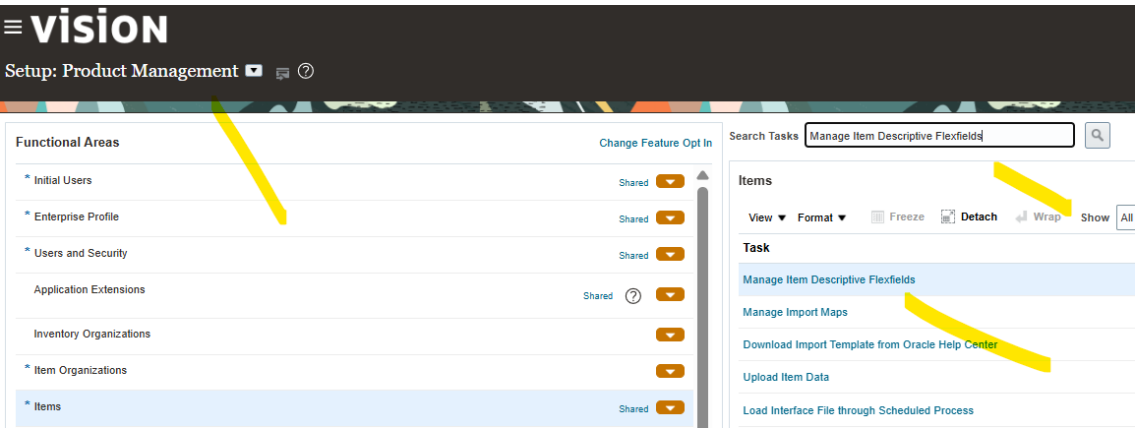
- Application Developer (ORA_FND_APPLICATION_DEVELOPER_JOB)
- Sync enabled developer identity (ORA_SYNC_ENABLED_DEVELOPER_ABSTRACT)

2. Set up flexfield.

Open SaaS and access menu as shown.

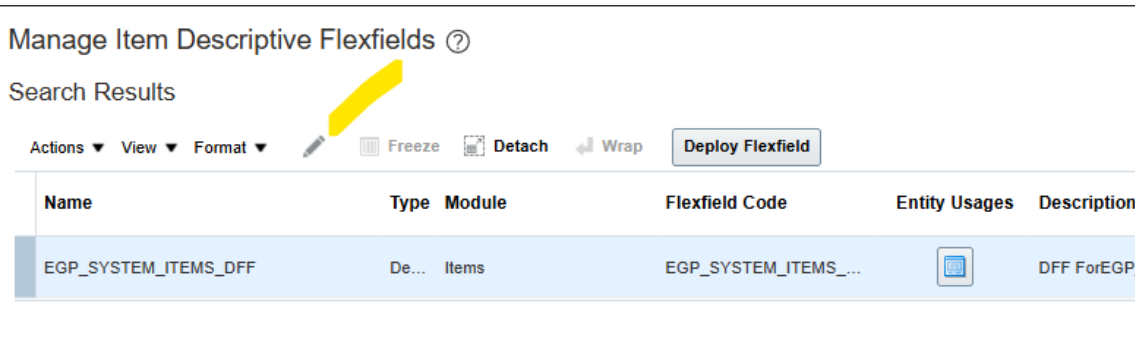


Search “Manage Item Descriptive Flexfields” in Product Management.



Name EGP_SYSTEM_ITEMS_DFF
Flexfield Code EGP_SYSTEM_ITEMS_DFF

Click on pencil to Edit.



Click on Plus icon to Create.

Manage Item Descriptive Flexfields [?](#)

Name EGP_SYSTEM_ITEMS_DFF

Flexfield Code EGP_SYSTEM_ITEMS_DFF

Description DFF For EGP_SYSTEM_ITEMS_DFF

Global Segments

Actions View Format + Freeze Detach Wrap

* Sequence	Name	Table Column	Value Set	Prompt
No data to display.				
Columns Hidden 9				

Context Segment

* Prompt Context

API Name

Value Set

Default Type

☐ Required

* Display Type Text Box

We have used this data form demo purposes (code provided relies on this exact setup).

Create Segment [?](#)

Flexfield Name EGP_SYSTEM_ITEMS_DFF

* Name EAN13_REDWOOD

* Code EAN13

* API Name ean13

Column Assignment

* Data Type Character

Validation

* Value Set 150Char

Value Set Description 150Char

Initial Default

Default Type

Display Properties

* Prompt EAN13_REDWOOD

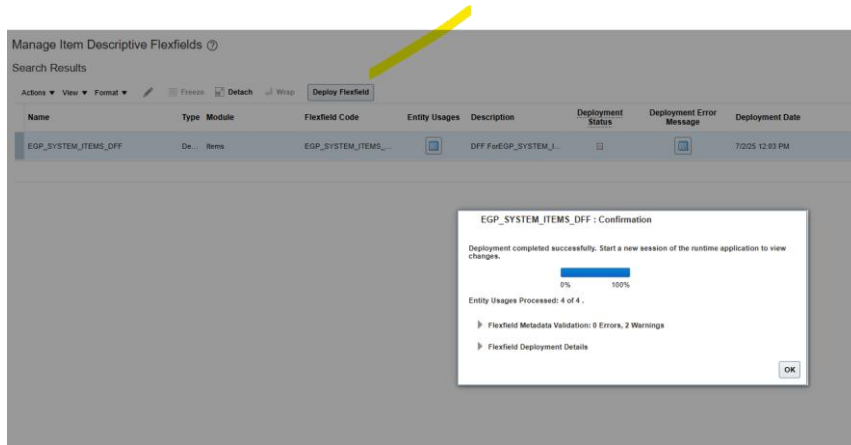
* Display Type Text Box

Display Size 30

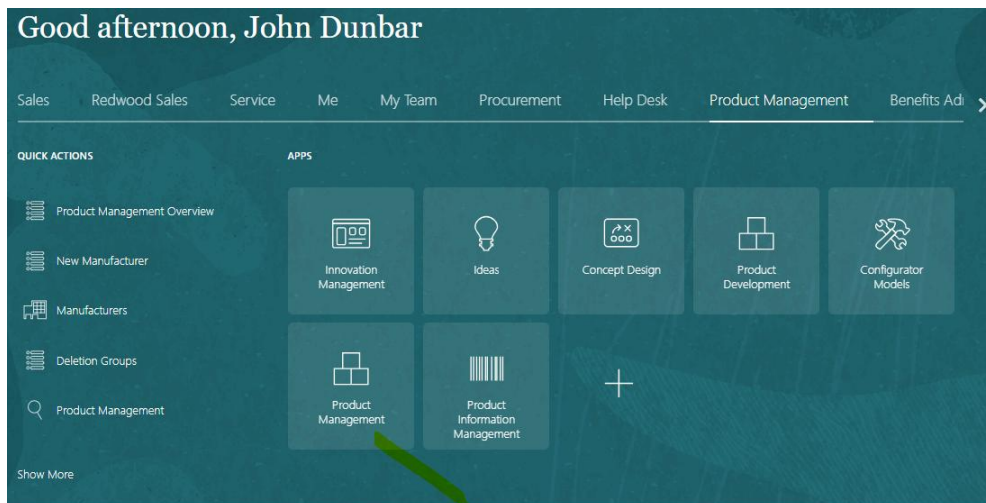
Display Height

☐ Read-only

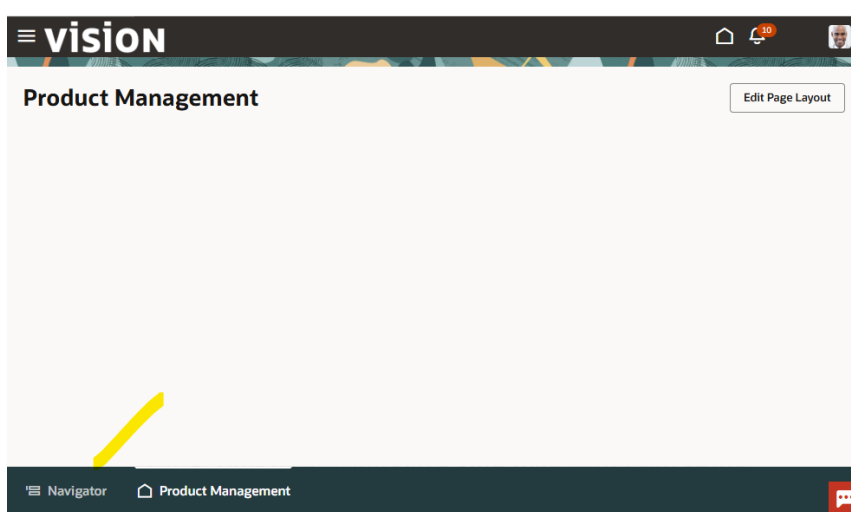
After “Save and close” x 2, run the “Deploy Flexfield” action.



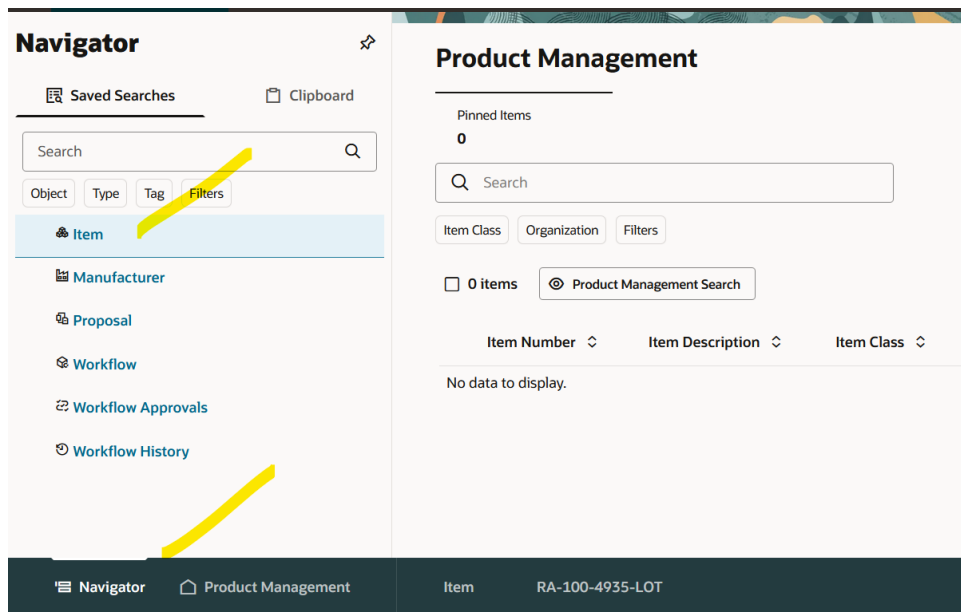
Re-login and search items.



Select “Navigator” below in the menu.



Feel free to select the item to play in your testing.

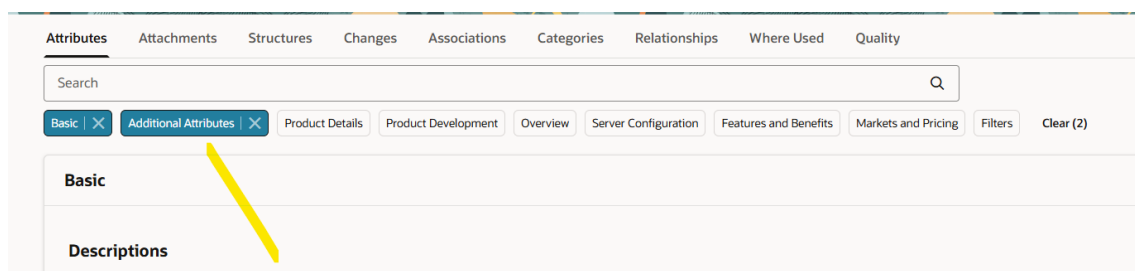


You will see a list of items.

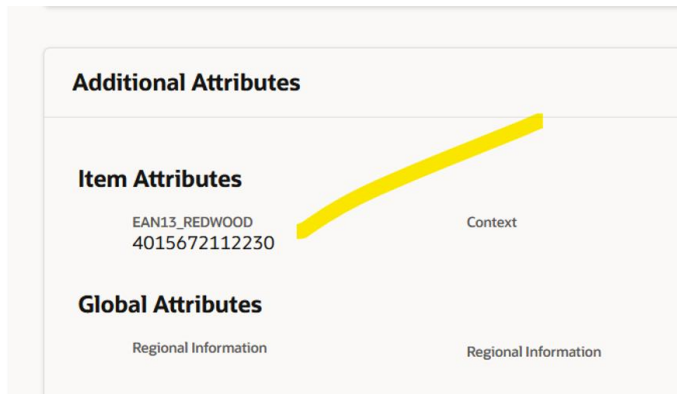
The screenshot shows the 'vision' Product Management Search interface. It includes a search bar, filters (Primary Unit of Measure, Item Class, Organization, Has structures, Has AML, Primary, Secondary, Tertiary, Processor, Speakers, Storage, Filters), and a table of items. A yellow arrow points to the 'Additional Attributes' tab in the 'Attributes' section.

Item Number	Item Description	Primary Unit of Measure	Item Class	Organization
<input type="checkbox"/> AS00102	Oracle Linux	Ea	Software	000
<input type="checkbox"/> AS00102	Oracle Linux	Ea	Software	001
<input type="checkbox"/> AS00102	Oracle Linux	Ea	Software	002
<input type="checkbox"/> AS00102	Oracle Linux	Ea	Software	003
<input type="checkbox"/> AS00102	Oracle Linux	Ea	Software	100

Select one item and click on “Additional Attributes” tab as shown.



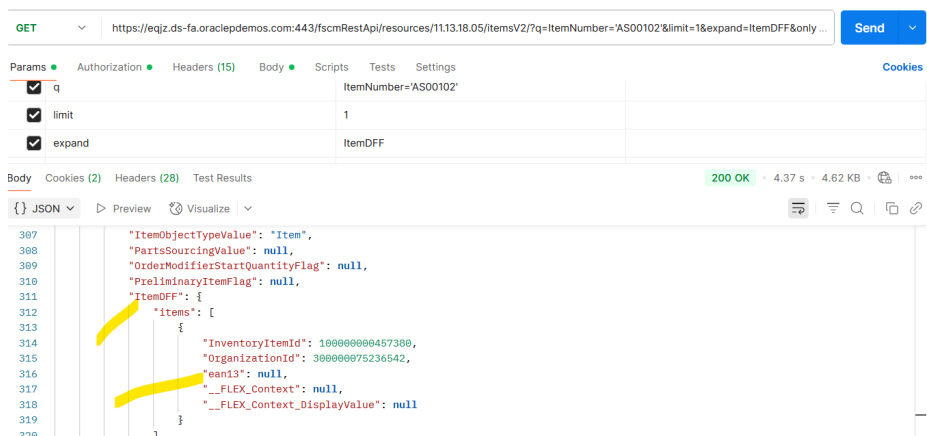
We should see the new flexfield.



Note.

Whatever the reason, in our testing the new flexfield does not show immediately, but we can see that our flexfield setup is ok using REST API with Postman.

<https://xxxx.ds-fa.oraclep demos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/?q=ItemNumber='AS00102'&limit=1&expand=ItemDFF&onlyData=true>



You could contact your SCM team to solve this issue (show flexfield).

We have solved updating the DFF data with Postman (see below in REST API section).

3. Highly recommended.

Investigate how these REST API work.

itemsV2.

Get Operation.

Example 1.

Query with item number and Organization.

```
https://zzzz-  
test.fa.em2.oraclecloud.com:443/fscmRestApi/resources/11.13.18.05//itemsV2?limit=50&q=ItemNumber='zzzzT' AND OrganizationCode='MASTER'&expand=all
```

Example 2.

Query with item flexfield DFF.

```
https://zzzz-  
test.fa.em2.oraclecloud.com:443/fscmRestApi/resources/11.13.18.05//itemsV2?limit=50&q=ItemDFF.ean13='8436039311787' AND OrganizationCode='MASTER'&expand=all
```

Note the following, in case you see error querying.

Steps to remove three (3) characters limits of Item Number from Search:

Fusion PIM: Three(3) Characters Limit to Enter Item Number On Advanced Search, SOAP Service & REST API (Doc ID 2651259.1)

Post Operation.

Update DFF using POST operation using REST API.

For this operation, first search item, as shown.

```
https://zzzz.ds-  
fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/?q=ItemNumber='AS00102'  
&limit=1&expand=ItemDFF&onlyData=true
```

GET https://eqjz.ds-fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/?q=ItemNur

Params	Authorization	Headers (15)	Body	Scripts	Tests	Settings
<input checked="" type="checkbox"/> q						ItemNumber='AS00102'
<input checked="" type="checkbox"/> limit						1
<input checked="" type="checkbox"/> expand						ItemDFF

Body Cookies (2) Headers (28) Test Results

{ } JSON Preview Visualize

```
1 {  
2   "items": [  
3     {  
4       "ItemId": 10000000457380,  
5       "OrganizationId": 300000075236542,  
6       "OrganizationCode": "050",  
7       "ItemClass": "Software",  
8       "Template": null,  
9       "ItemNumber": "AS00102",  
10      "ItemDescription": "Oracle Linux",
```

Notice no data in flexfield yet.

GET <https://eqjz.ds-fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2?q=ItemNumber='AS00102'&limit=1&expand=ItemDFF&only...> Send

Params Authorization Headers (15) Body Scripts Tests Settings Cookies

q ItemNumber='AS00102'

limit 1

expand ItemDFF

Body Cookies (2) Headers (28) Test Results 200 OK 4.37 s 4.62 KB

JSON Preview Visualize

```

307      "ItemObjectTypeValue": "Item",
308      "PartsSourcingValue": null,
309      "OrderModifierStartQuantityFlag": null,
310      "PreliminaryItemFlag": null,
311      "ItemDFF": {
312        "items": [
313          {
314            "InventoryItemId": 100000000457380,
315            "OrganizationId": 300000075236542,
316            "ean13": null,
317            "__FLEX_Context": null,
318            "__FLEX_Context_DisplayValue": null
319          }
320        ]
321      }
  
```

Notice the ids for item and inventory organization.

```

{
  "items": [
    {
      "ItemId": 100000000457380,
      "OrganizationId": 300000075236542,
      "OrganizationCode": "050",
    }
  ]
}
  
```

Notice the long string in next screen, you will need it below to run the Post operation.

GET <https://eqjz.ds-fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2?q=ItemNumber='AS00102'&limit=1&expand=ItemDFF&only...> Send

Params Authorization Headers (15) Body Scripts Tests Settings Cookies

q ItemNumber='AS00102'

limit 1

expand ItemDFF

Body Cookies (2) Headers (28) Test Results 200 OK 4.37 s 4.62 KB

JSON Preview Visualize

```

322      "hasMore": false,
323      "limit": 1,
324      "offset": 0,
325      "links": [
326        {
327          "rel": "self",
328          "href": "https://eqjz.ds-fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/000400000000EACED000577080005AF310813AA40000000EACED00057708000110D935EAC4BE0000000EACED0005770800005AF310813AA40000000EACED00057708000110D935EAC4BE/child/ItemDFF",
329          "name": "ItemDFF",
330          "kind": "collection"
331        }
332      ]
333    }
  
```

Run as shown.

```

https://xxxx.ds-
fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/000400000000EACED000577
0800005AF310813AA40000000EACED00057708000110D935EAC4BE0000000EACED0005770800005AF310813A
A40000000EACED00057708000110D935EAC4BE/child/ItemDFF
  
```

Notice after running with the payload shown that we have updated our flexfield.

https://fa-exgr-saasfademo1.ds-fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/00040000000EACED000577080...

POST https://eqjz.ds-fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/00040000000EACED0005770800005AF310813AA400000...

Params Authorization Headers (15) Body Scripts Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "InventoryItemId": 100000000457380,
3   "OrganizationId": 300000075236542,
4   "ean13": "4015672112230"
5 }

```

Body Cookies (2) Headers (29) Test Results 201 Created 3.89 s 2.4

{ } JSON Preview Visualize

```

1 {
2   "InventoryItemId": 100000000457380,
3   "OrganizationId": 300000075236542,
4   "ean13": "4015672112230",
5   "__FLEX_Context": null,
6   "__FLEX_Context_DisplayValue": null,
7   "links": [
8     {
9       "rel": "self",
10      "href": "https://eqjz.ds-fa.oraclepdemos.com:443/fscmRestApi/resources/11.13.18.05/itemsV2/"

```

Note: the EAN13 code (4015672112230) used to update code is a real one and matches our real barcode for testing. You could use yours.

Provided our photo-picture for your testing.



Stocks Inquiry.

<https://docs.oracle.com/en/cloud/saas/supply-chain-and-manufacturing/24b/fasrp/op-itemstockinginquiries-get.html>

You can use these queryable attributes to filter this collection resource using the q query parameter:

- Description; string; Value that identifies the description of the item.
- InventoryItemId; integer; Value that uniquely identifies the inventory item.
- ItemNumber; string; Number that identifies the inventory item.
- LastCountDate; string; Date when the item was last counted for a location during cycle counting.
- LastReplenishedDate; string; Date when the item was last replenished for a location during par counting.
- Locator; string; Physical area in the subinventory such as a row, aisle, bin, or shelf, that stores the item.
- LocatorId; integer; Value that uniquely identifies the locator.
- LotNumber; string; Number that identifies a lot.
- OrganizationCode; string; Abbreviation that identifies the organization.
- OrganizationId; integer; Value that uniquely identifies the organization.

- ParLocation; integer; Value that indicates if the on-hand quantity is associated with a subinventory that's enabled for a par location inventory.
- PrimaryImageURL; string; Value that provides a reference to the image of the item for which the count sequence is generated.
- PrimaryOnhandQuantity; number; Quantity of the item in the primary unit of measure.
- PrimaryUOMCode; string; Abbreviation that identifies the primary unit of measure.
- PrimaryUnitOfMeasure; string; Primary unit of measure of the item.
- SerialNumber; string; Serial number of the item in an inventory lot.
- StockingOnhandQuantity; number; Value that indicates the on-hand quantity for an item in the stocking unit of measure of the subinventory.
- StockingUOMCode; string; Value that indicates the unit of measure code in which the items are stocked in a subinventory.
- StockingUnitOfMeasure; string; Value that indicates the unit of measure in which the items are stocked in a subinventory.
- SubinventoryCode; string; Abbreviation that identifies the subinventory in the organization. This value references a physical area or a logical grouping of items such as a storeroom or receiving dock.

Inventory Organizations.

<https://docs.oracle.com/en/cloud/saas/supply-chain-and-manufacturing/24b/fasrp/op-inventoryorganizations-get.html>

The screenshot shows the 'REST API for Oracle Fusion Cloud SCM' documentation page. The left sidebar contains a navigation menu with categories like Feature States, Inventory Organizations, Parameters, Plant Parameters, Item Unit of Measure Conversions, Packaging Strings, Production Calendar Shift Exceptions, Project Attribute Defaults, Project Type Cost Exclusions, Schedule Exceptions, and Schedules. The main content area is titled 'Get all organizations' and includes a 'GET' button, the endpoint path '/fscmRestApi/resources/11.13.18.05/inventoryOrganizations', and a 'Request' section with 'Query Parameters'. The query parameters listed are: expand: string, fields: string, finder: string, limit: integer, links: string, and offset: integer.

<https://fa-xxxx-saasfademo1.ds-fa.oraclepdemos.com:443//fscmRestApi/resources/11.13.18.05/inventoryOrganizations?onlyData=true&q=OrganizationCode='000'>

We can see here which is the Master Organization, we will need it below.

GET https://fa-esll-saasfademo1.ds-fa.oraclepdemos.com:443//fscmRestApi/resources/11.13.18.05/inventoryOrganizations?onlyData=true&q=Organiza...

Params Authorization Headers (14) Body Scripts Tests Settings

Query Params

<input checked="" type="checkbox"/> Key	Value	Description
---	-------	-------------

Body Cookies Headers (28) Test Results 200 OK 1.36 s 1.74 K

{ } JSON Preview Visualize

```
11 "ProfitCenterBusinessUnitId": null,
12 "ProfitCenterBusinessUnitName": null,
13 "Status": "Active",
14 "LocationId": 300000047013150,
15 "LocationCode": "Headquarters",
16 "InventoryFlag": false,
17 "ManufacturingPlantFlag": false,
18 "ContractManufacturingFlag": false,
19 "MaintenanceEnabledFlag": false,
20 "MasterOrganizationId": 300000047274441,
21 "MasterOrganizationCode": "000",
22 "MasterOrganizationName": "Operations",
23 "IntegratedSystemType": null,
```

master Aa .ab * 1 of 3 ↑ ↓

Detailed steps followed.

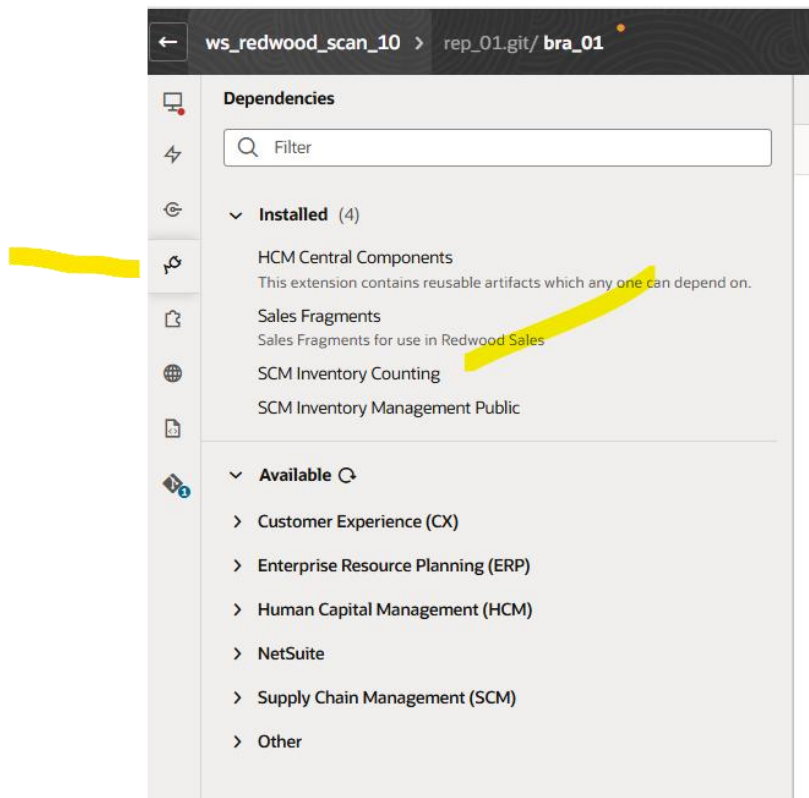
1. Project creation.

To not repeat...follow steps from Part I: Redwood_CookBook 2026 SCM ITEM SCANNER.pdf

We will show in this document, only the steps after been able to show “xx” component.

Summary: we created an Extension using “Welcome template page” and added a “**Barcode Scanner**” component.

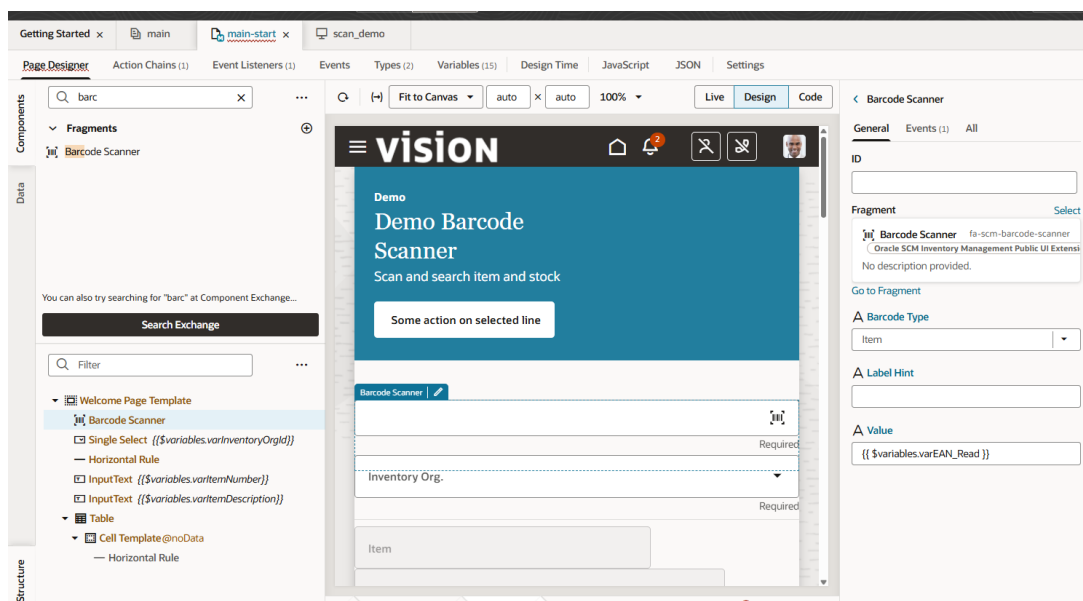
2. Make sure you have these dependencies.



3. We need the following variables.

Variables		Extension Read	Extension W
A	backgroundColor	—	—
A	illustrationBackground	—	—
A	illustrationForeground	—	—
A	imageStretch	—	—
SDP	inventoryOrganizationsLOVListSDP	—	—
SDP	itemStockingInquiriesListSDP	—	—
A	mobileImage	—	—
#	varCount	—	—
A	varEAN_Read	✓	✓
A	varInventoryOrgId	✓	✓
A	varItemDescription	—	—
A	varItemId	—	—
A	varItemNumber	✓	✓
A	varMasterOrgCode	—	—
A	varQuery	—	—

Assign the variable to “Barcode Scanner” component as shown.



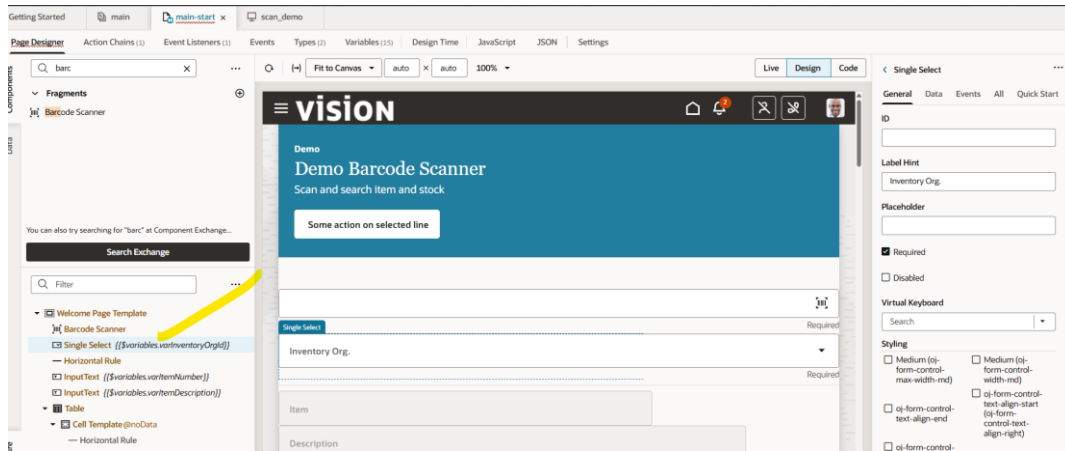
And setup in default value for variable with Master Inventory Organization (see above in REST API to find how to get it).

Variables		Extension Read	Extension Write
A	backgroundColor	—	—
A	illustrationBackground	—	—
A	illustrationForeground	—	—
A	imageStretch	—	—
SDP	inventoryOrganizationsLOVListSDP	—	—
SDP	itemStockingInquiriesListSDP	—	—
A	mobileImage	—	—
#	varCount	—	—
A	varEAN_Read	✓	✓
A	varInventoryOrgId	✓	✓
A	varItemDescription	—	—
A	varItemId	—	—
A	varItemNumber	✓	✓
A	varMasterOrgCode	—	—
A	varQuery	—	—

varMasterOrgCode	Type *	String	Create
	Label		
	Description	varMasterOrgCode	
	Access for Application Extensions *	No Access	Read Only
	Input Parameter	Disabled	Enabled
	Default Value	000	
	Dirty Data Behavior	None	
	Persisted	None	

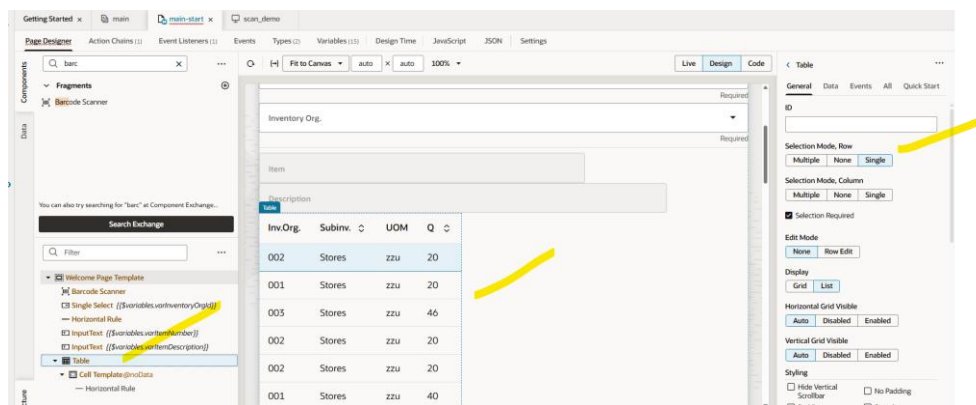
4. Add "Select single" for Inventory organization.

See in REST API section.

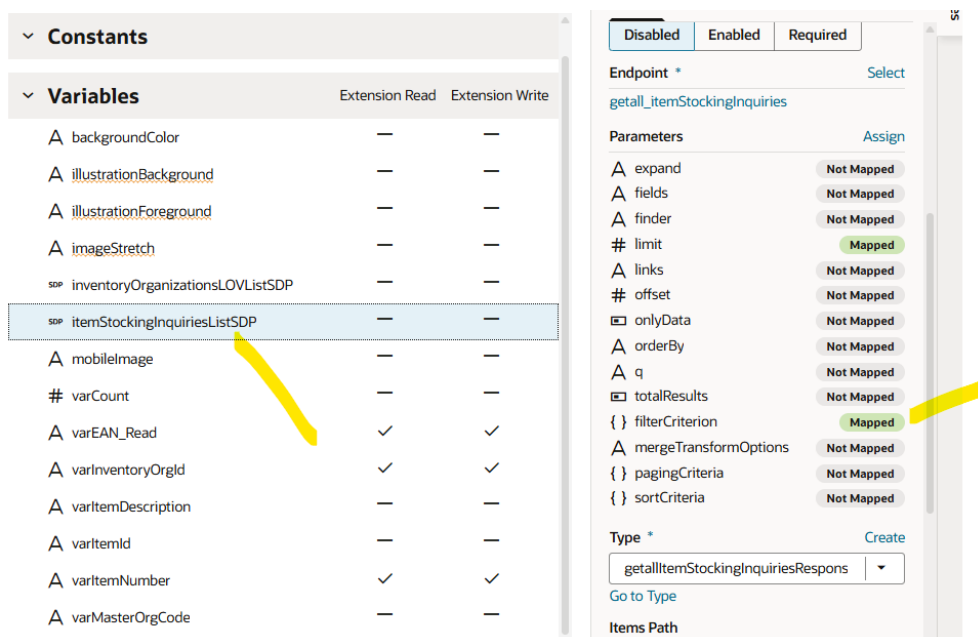


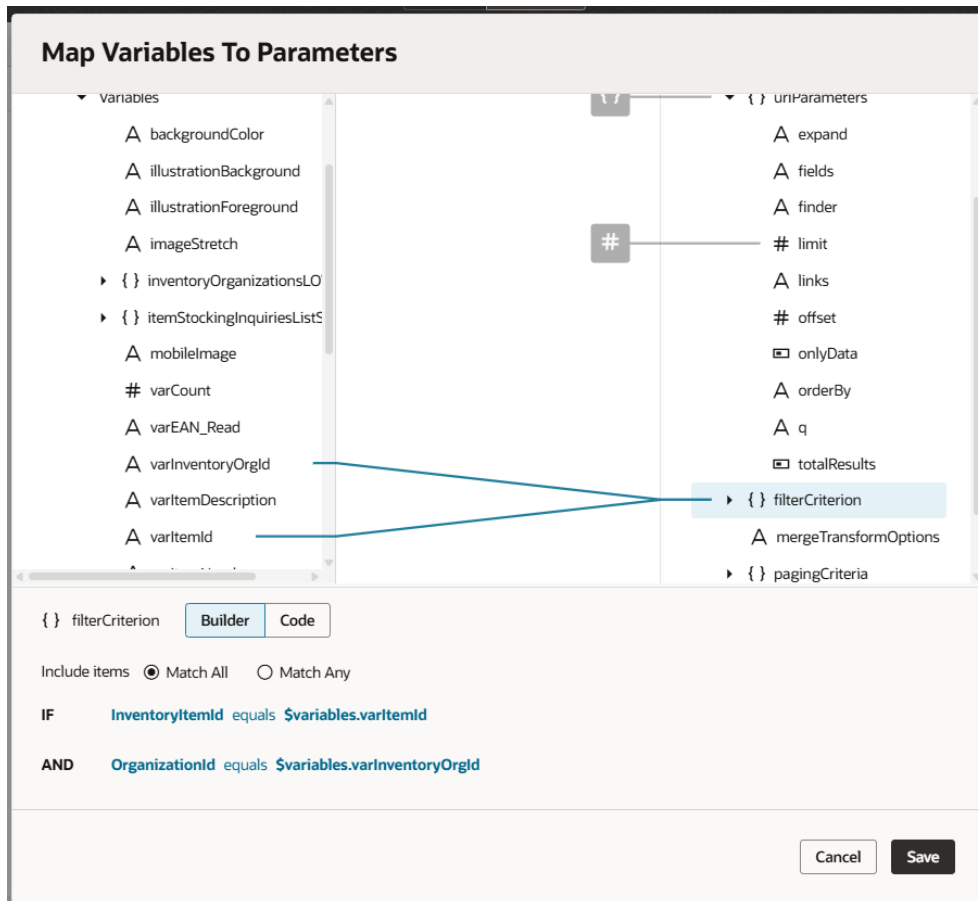
5. Add "Table" component for stocks.

See in REST API section.

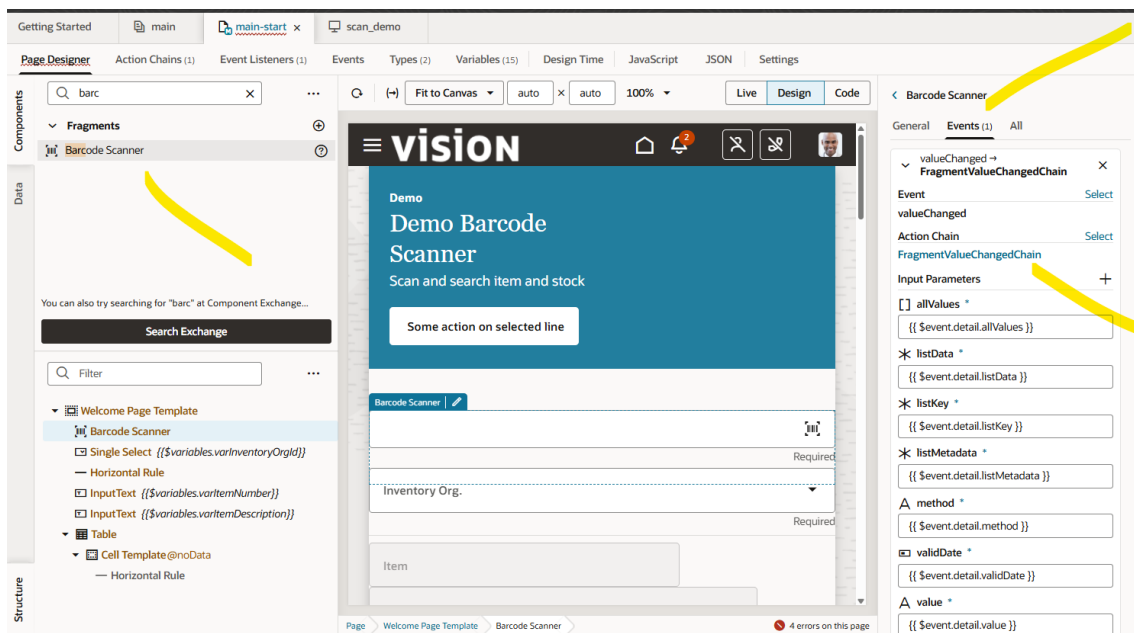


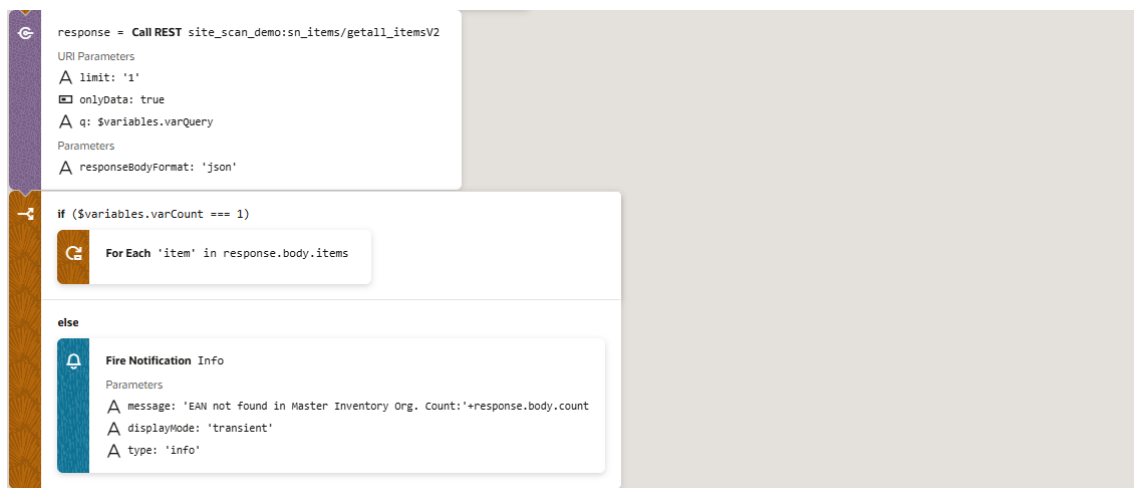
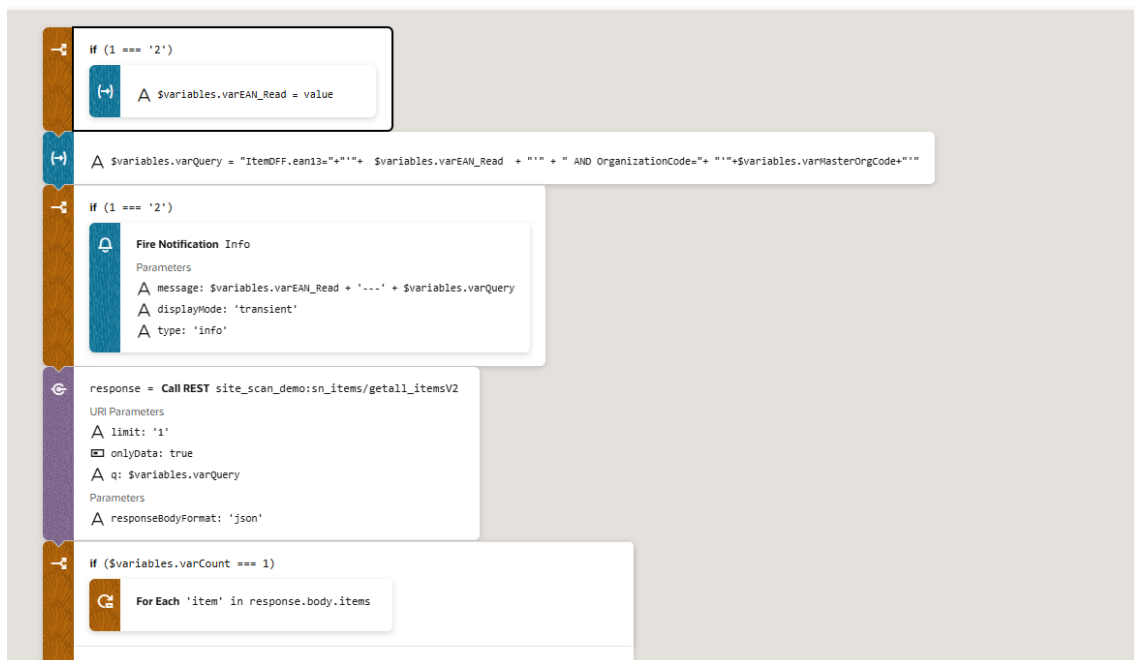
6. Set up filter for resulting variable SDP.





7. Add "Action chain" to search information, after the user scan EAN13 or type directly.





```

define([
  'vb/action/actionChain',
  'vb/action/actions',
  'vb/action/actionUtils',
], (
  ActionChain,
  Actions,
  ActionUtils
) => {
  'use strict';

  class FragmentValueChangedChain extends ActionChain {
    /**
     * @param {Object} context
     * @param {Object} params
     * @param {string} params.value
     * @param {string} params.method
     * @param {any} params.listKey
     * @param {any} params.listData
     * @param {any} params.listMetadata
     * @param {object[]} params.allValues
     * @param {boolean} params.validate
     */
    async run(context, { value, method, listKey, listData, listMetadata, allValues, validate }) {
      const { $page, $flow, $application, $base, $extension, $constants, $variables } = context;

      if (1 === '2') {
        $variables.varEAN_Read = value;
      }

      $variables.varQuery = "ItemDFF.ean13="+""+ $variables.varEAN_Read + "" + " AND OrganizationCode="+
        ""+$variables.varMasterOrgCode+"";

      if (1 === '2') {

```

```

        await Actions.fireNotificationEvent(context, {
            summary: 'Info',
            type: 'info',
            displayMode: 'transient',
            message: $variables.varEAN_Read + '---' + $variables.varQuery,
        });
    }

    const response = await Actions.callRest(context, {
        endpoint: 'site_scan_demo:sn_items/getall_itemsV2',
        responseBodyFormat: 'json',
        uriParams: {
            q: $variables.varQuery,
            onlyData: true,
            limit: '1',
        },
    }, { id: 'rest1' });

    $variables.varCount = response.body.count;

    if ($variables.varCount === 1) {

        const results = await ActionUtils.forEach(response.body.items, async (item, index) => {

            $variables.varItemId = item.ItemId;
            $variables.varItemDescription = item.ItemDescription;
            $variables.varItemNumber = item.ItemNumber;
        }, { mode: 'serial' });
    }
    else{

        await Actions.fireNotificationEvent(context, {
            summary: 'Info',
            message: 'EAN not found in Master Inventory Org. Count:'+response.body.count,
            displayMode: 'transient',
            type: 'info',
        });
    }
}

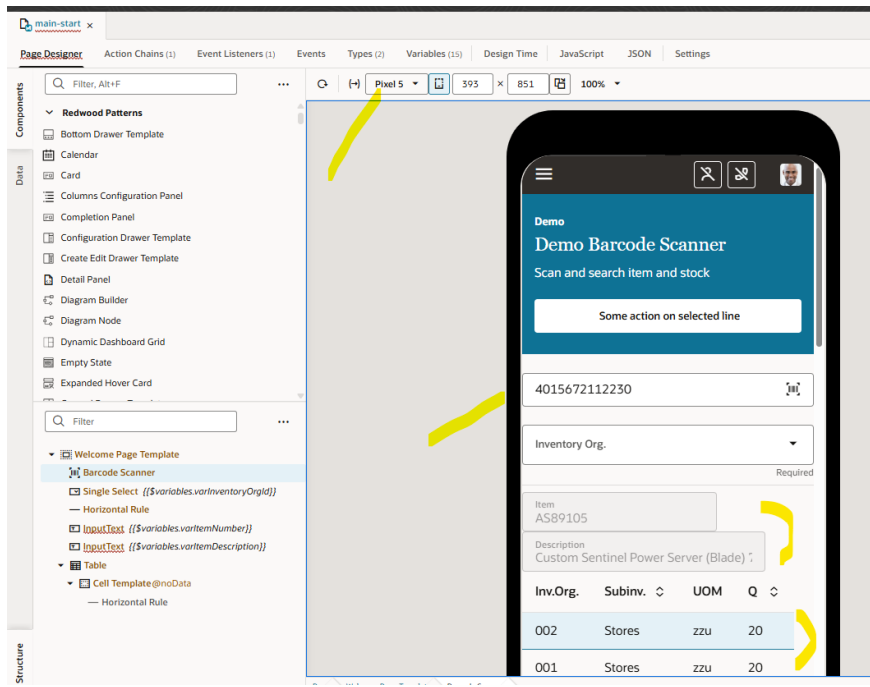
return FragmentValueChangedChain;
});

```

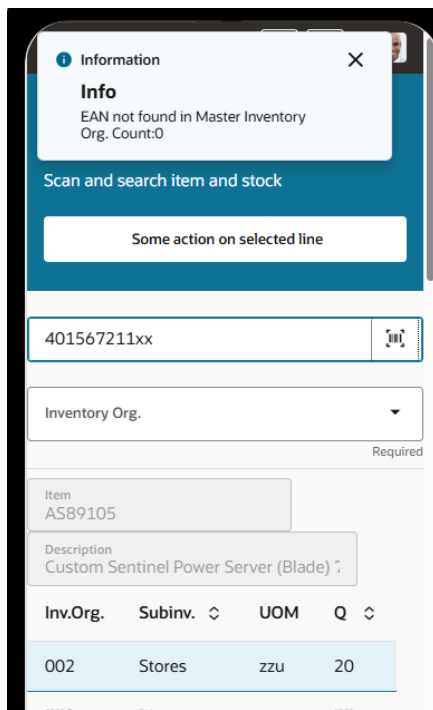
Testing-evidence.

a) Test from Design mode.

We will type directly the barcode.

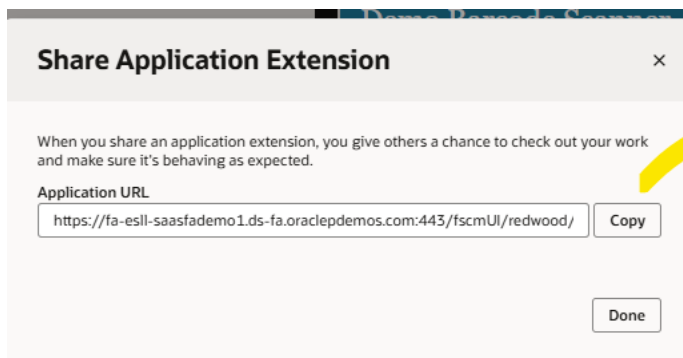
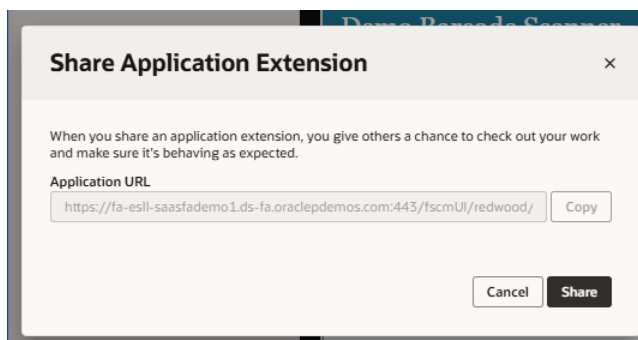
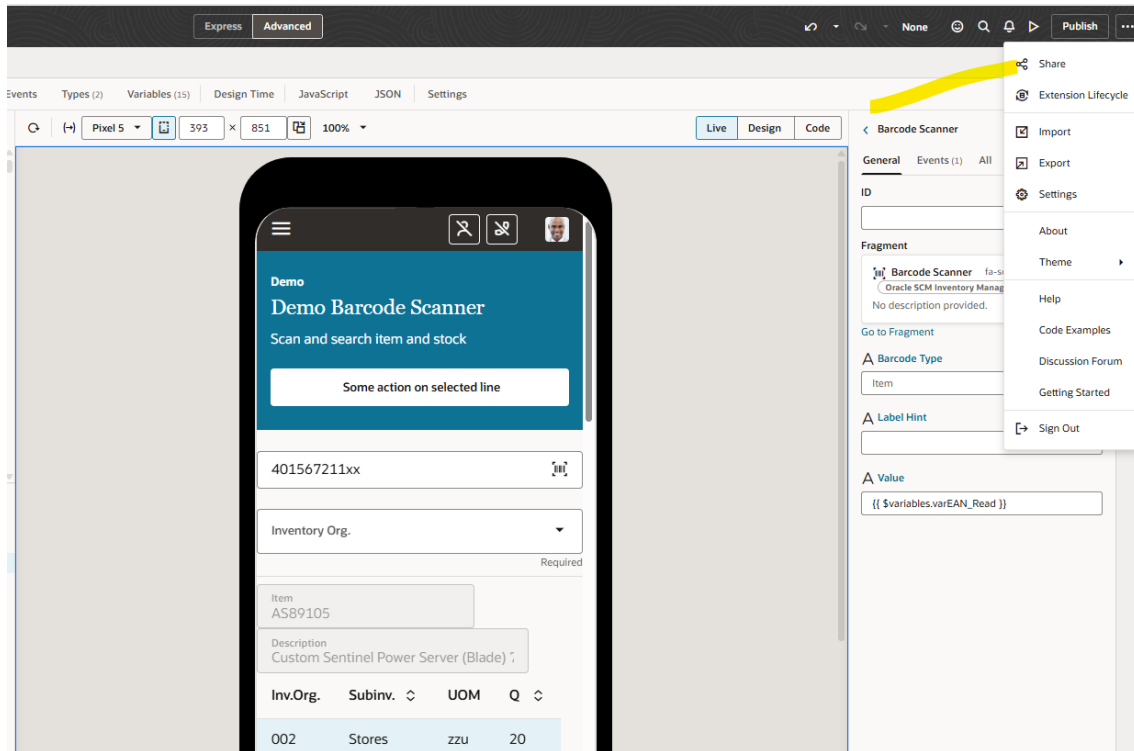


Testing when item is not found.



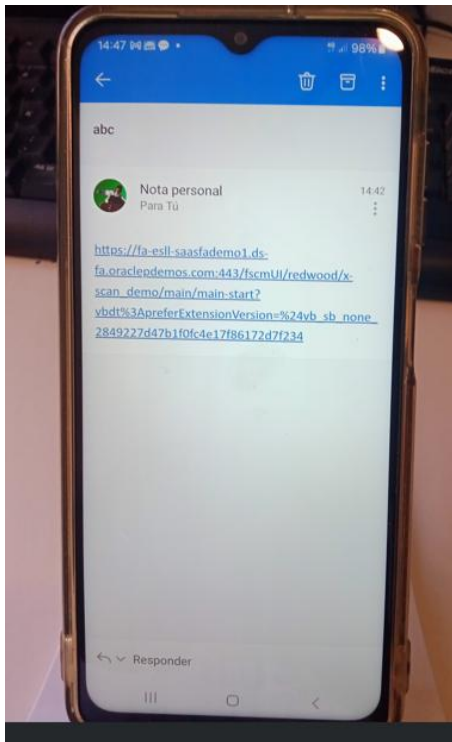
b) Test with real device

- Share URL to your device, mail for my demo.

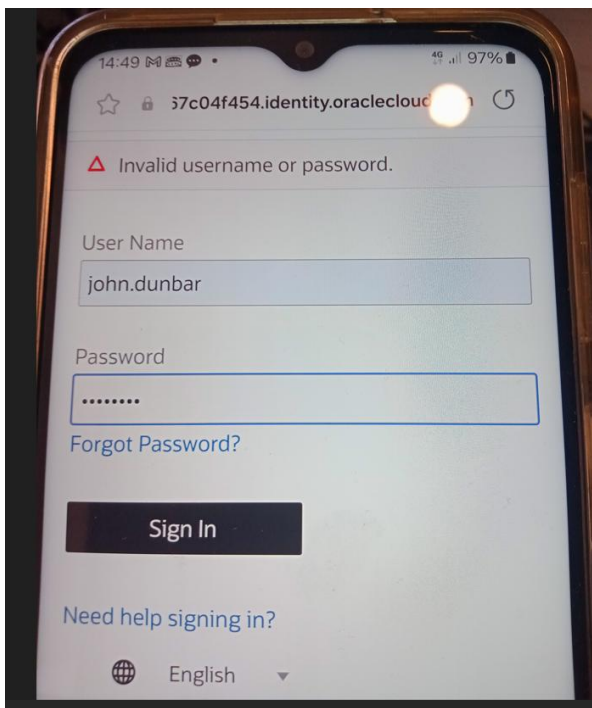


`https://fa-qqqg-saasfademo1.ds-fa.oraclepdemos.com:443/fscmUI/redwood/x-scan_demo/main/main-start?vbdt%3ApreferExtensionVersion=%24vb_sb_none_2849227d47b1f0fc4e17f86172d7f234`

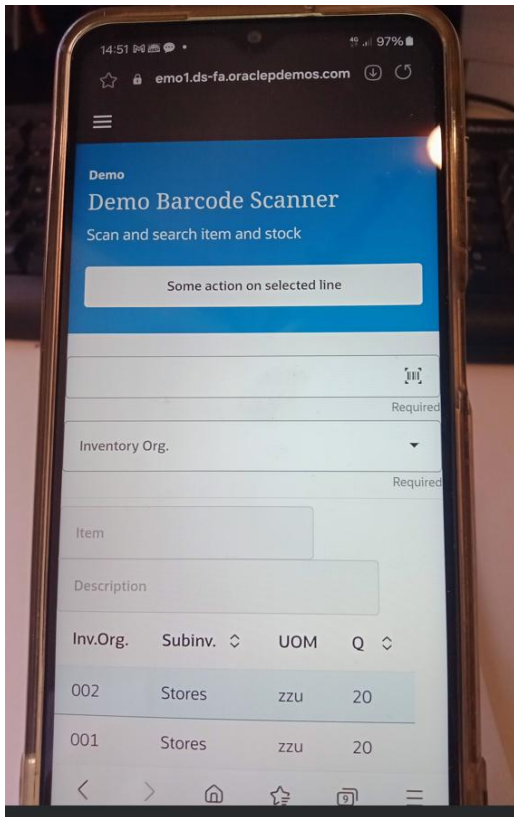
- Open URL from device



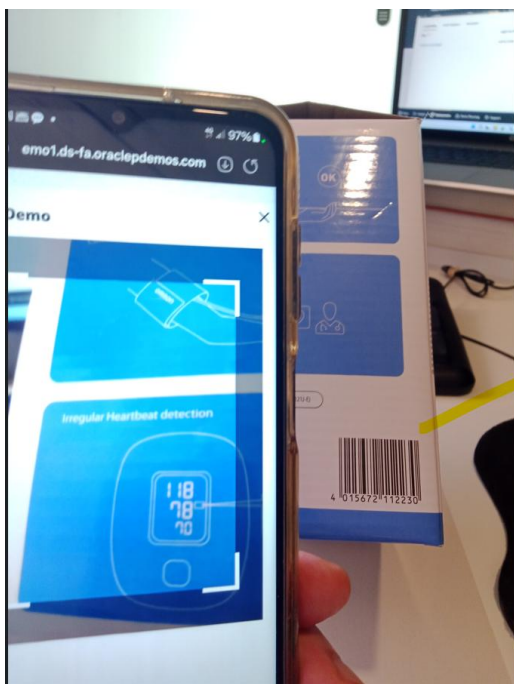
- Enter credentials.



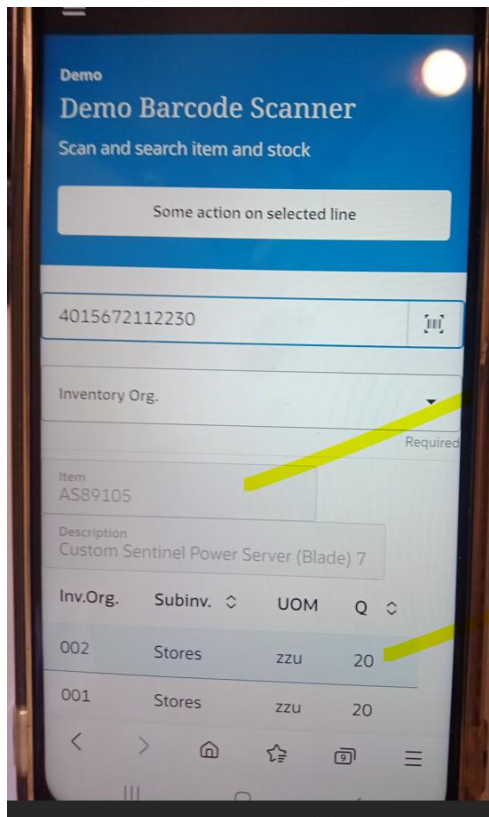
- Before any action



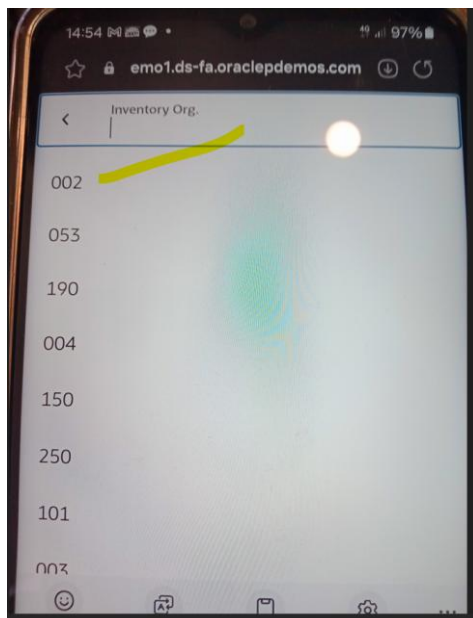
- We will focus on real barcode, that matches our flexfield (ean13)



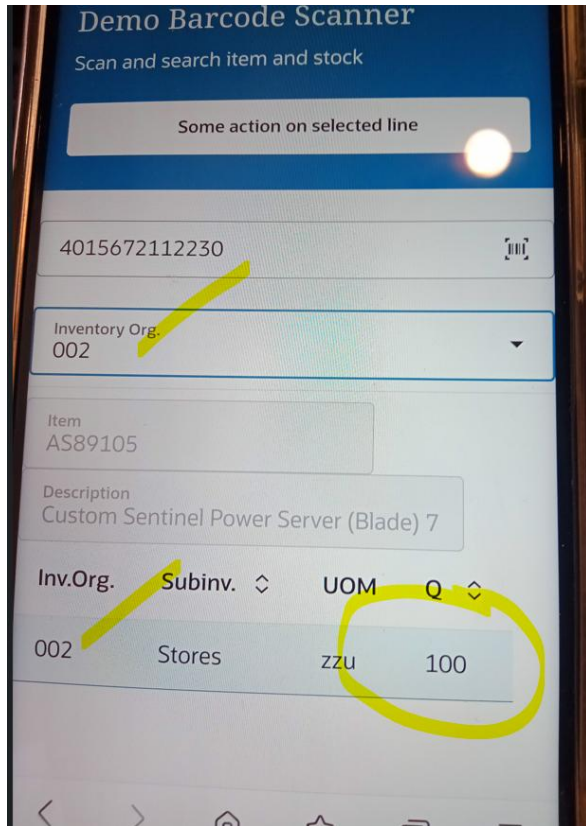
- After scan is ok, item and stock is found.



- After using list of values to select Inventory Organization



- We filter accordingly to limit the data retrieved.



Conclusion.

We have been able to develop a COMPLETE basic application to scan a EAN13 item using a mobile device, showing item data and stock data.

Redwood Visual Builder responsive features will work for us “out of the box” to get the job done.

Technical

Code and documentation in: https://github.com/juanjesusmontero/oracle_redwood_cookbook

Documents:

Redwood_CookBook 2026 SCM ITEM SCANNER.pdf

Redwood_CookBook 2026 SCM ITEM SCANNER Part II.pdf

Code example	Comments
Redwood_CookBook	Code used in extension for the demo.
2026_workspace_dev4_scan_v01.zip	Import as workspace.