

SCM. Master-detail-detail Redwood page for Sales Orders with Hash key.

Redwood Cookbook 2026. Advanced chapter.

Introduction/Goal

Oracle fusion standard REST APIs sometimes use what is called “Hash keys”.

Without knowledge of this feature, you will not be able to use many REST APIs, for example for SCM Sales Orders and a few others.

Explanation from the AI:

In the context of Oracle REST APIs, hash codes are used to uniquely identify resources, particularly when dealing with composite keys. Instead of relying on the client to generate these hash keys, it's recommended to retrieve them by querying the relevant resource collection.

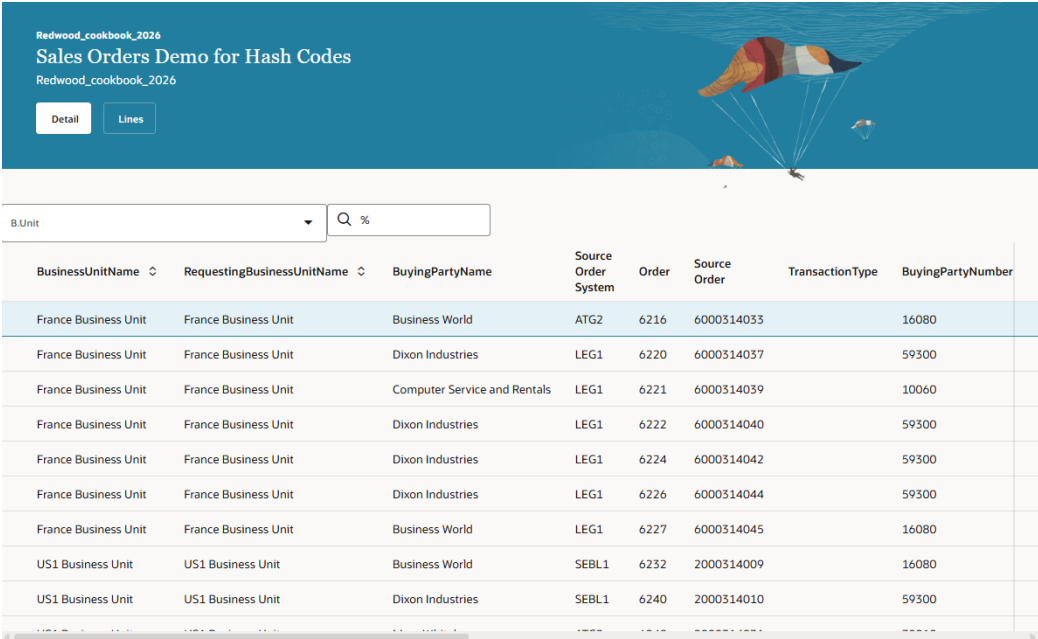
From us:

The hash codes are “special” filters for some REST APIs, usually to access child nodes.

To deal with them you have to “ask” the parent node which is the Hash code for a specific child.

We will build a page with 3 levels (master-detail-detail) using hash codes for the third level to show how these work.

SCM Sales Orders will be the chosen REST APIs for this exercise.



BusinessUnitName	RequestingBusinessUnitName	BuyingPartyName	Source Order System	Order	Source Order	TransactionType	BuyingPartyNumber
France Business Unit	France Business Unit	Business World	ATG2	6216	6000314033		16080
France Business Unit	France Business Unit	Dixon Industries	LEG1	6220	6000314037		59300
France Business Unit	France Business Unit	Computer Service and Rentals	LEG1	6221	6000314039		10060
France Business Unit	France Business Unit	Dixon Industries	LEG1	6222	6000314040		59300
France Business Unit	France Business Unit	Dixon Industries	LEG1	6224	6000314042		59300
France Business Unit	France Business Unit	Dixon Industries	LEG1	6226	6000314044		59300
France Business Unit	France Business Unit	Business World	LEG1	6227	6000314045		16080
US1 Business Unit	US1 Business Unit	Business World	SEBL1	6232	2000314009		16080
US1 Business Unit	US1 Business Unit	Dixon Industries	SEBL1	6240	2000314010		59300

DISCLAIMER: this demo is for investigation, use at your discretion, no guarantee provided.

REST API investigation.

<https://docs.oracle.com/en/cloud/saas/supply-chain-and-manufacturing/25c/fasrp/op-salesordersfororderhub-get.html>

The following REST operation (GET all sales orders) will be used for Master.

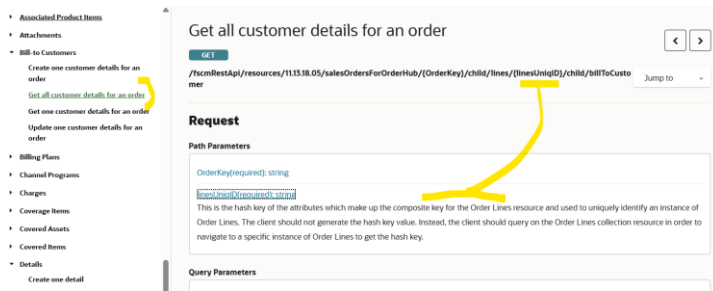
The screenshot shows a REST API client interface. On the left is a sidebar with a tree view of API endpoints. The main panel displays the details for the 'Get all sales orders' endpoint. The endpoint is a GET request to `/fscmRestApi/resources/11.13.18.05/salesOrdersForOrderHub`. Under the 'Request' section, there is a 'Query Parameters' field with the value `expand: string`.

The following REST (GET all order lines) operation ABOVE will be used for first detail.

The screenshot shows the REST API client interface for the 'Get all order lines' endpoint. The endpoint is a GET request to `/fscmRestApi/resources/11.13.18.05/salesOrdersForOrderHub/{OrderKey}/child/lines`. Under the 'Request' section, there are two parameter lists: 'Path Parameters' with `OrderKey(required): string`, and 'Query Parameters' with `dependency: string`, `expand: string`, `fields: string`, `finder: string`, `limit: integer`, and `links: string`. A yellow arrow points from the 'Get all order lines' option in the sidebar to the endpoint title.

The following REST (GET all customer details for an order) operation BELOW will be used for second level detail.

Here we see that second parameter is “special”, it is something that cannot be queried directly, and do not exist in the SaaS database.



Helper OIC integration to get the Hash Code.

Reference doc.

<https://community.oracle.com/customerconnect/discussion/850275/how-to-get-hash-key-for-rest-api-input>

This integration has been built for our convenience; it is easier for us to deal with these types of payloads in OIC.

Feel free to investigate how to replicate this approach to finding the Hash keys in your Visual Builder application.

In the OIC integration we will have:

- 1) These connections.

Configuration Editor - XX_SCM_HASH_FOR_VB PACKAGE		
<div> <div>Connections</div> <div>Edit or replace to configure connections</div> </div> <div> <div>Connections</div> <div>Activation</div> </div>		
Name	Usage in this package	Status
ERP_Cloud_Connection_REST	1 Integration	Configured
XX_DEMO_REST	1 Integration	Configured

<

XX_DEMO_REST

Configured

Role
Trigger

Identifier
XX_DEMO_REST

Updated on
Mar 19, 2025, 10:10:43 AM CET

Used in
25 integrations


Security

Security policy
OAuth 2.0 Or Basic Authentication

Access type

☒ Public gateway
Connect to endpoints using the internet.

☐ Private endpoint
Connect to endpoints using your private network.


View Request Parameters
REST trigger

Operation Name:
default


Resource URI:
/

HTTP Method:
GET

Specify Query Parameters

<input type="checkbox"/>	Name	Data Type
<input type="checkbox"/>	pOrderKey	string
<input type="checkbox"/>	pLineId	string

3) Response parameters.


View Response
REST trigger

Enter Sample JSON

```
{ "HashCode" : "xx", "CompleteURL" : "yy" }
```

Close

Select the multipart attachment processing options
☐ Response is multipart with payload
☐ Multipart response is of type multipart/form-data with HTML form payload

Select the response payload format
JSON Sample

Dropping file is disabled

--OR-- enter sample JSON
<<< inline >>>

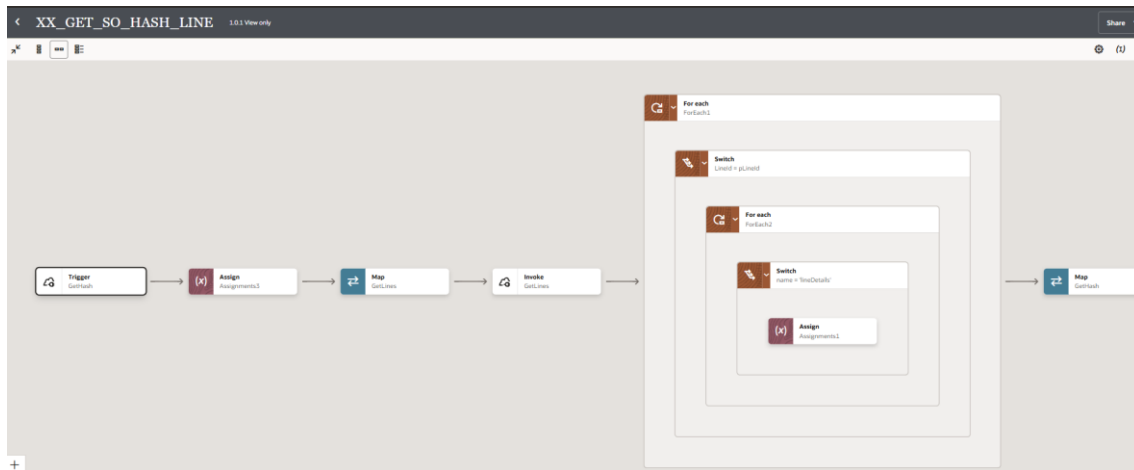
Element
response-wrapper

What is the media-type of Response Body? (Accept Header)
JSON

Media Type

Close Previous Next

4) General overview.



5) Details of the integration.

a) REST API call.



View Request Parameters

REST invoke

Operation Name:
default

Resource URI:
/fscmRestApi/resources/11.13.18.05/salesOrdersForOrderHub/{id}/child/lines/

HTTP Method:
GET

Specify Query Parameters

<input type="checkbox"/>	Name	Data Type
--------------------------	------	-----------

No data to display.

Template Parameters

Displays the template parameters in the relative resource URI. Template parameters are determined by details you specified on the Basic Info page and cannot be edited.

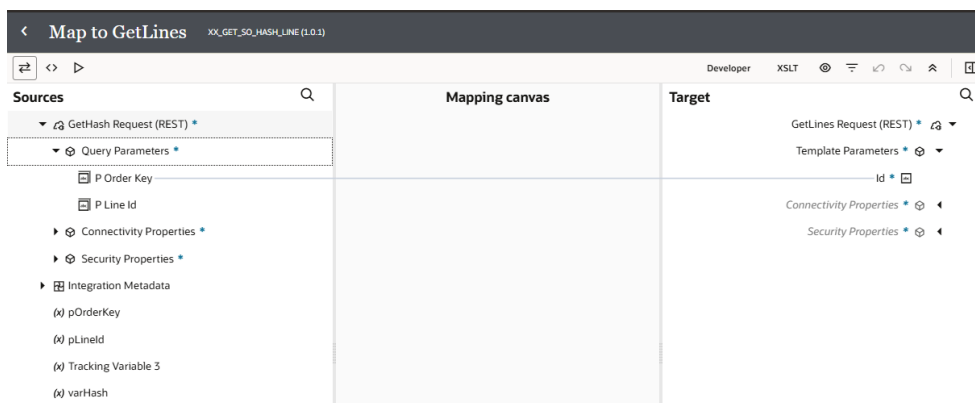
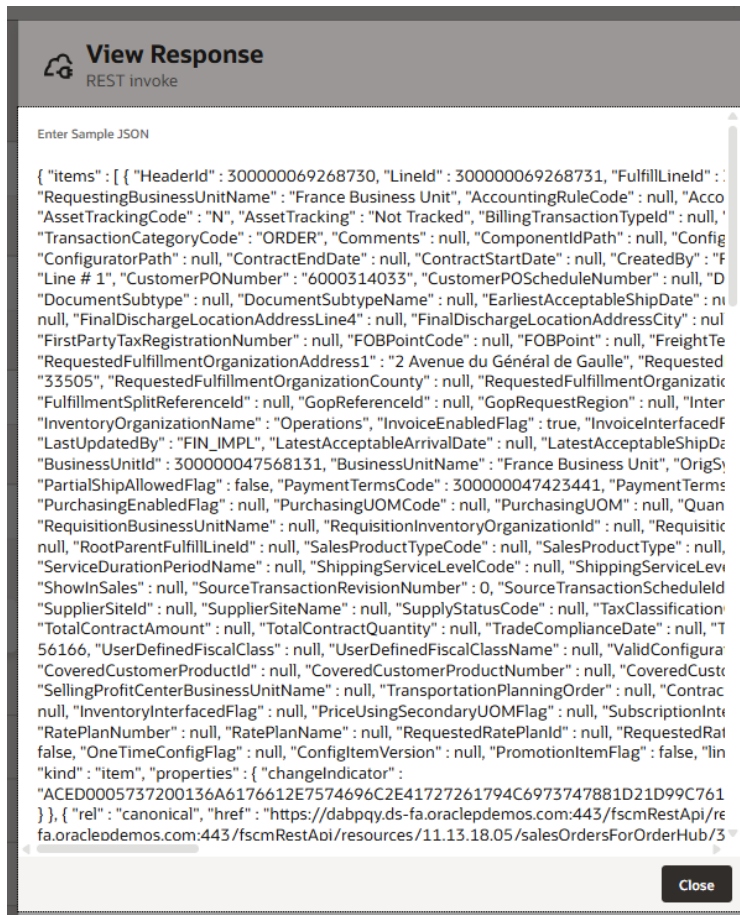
Name
id

string

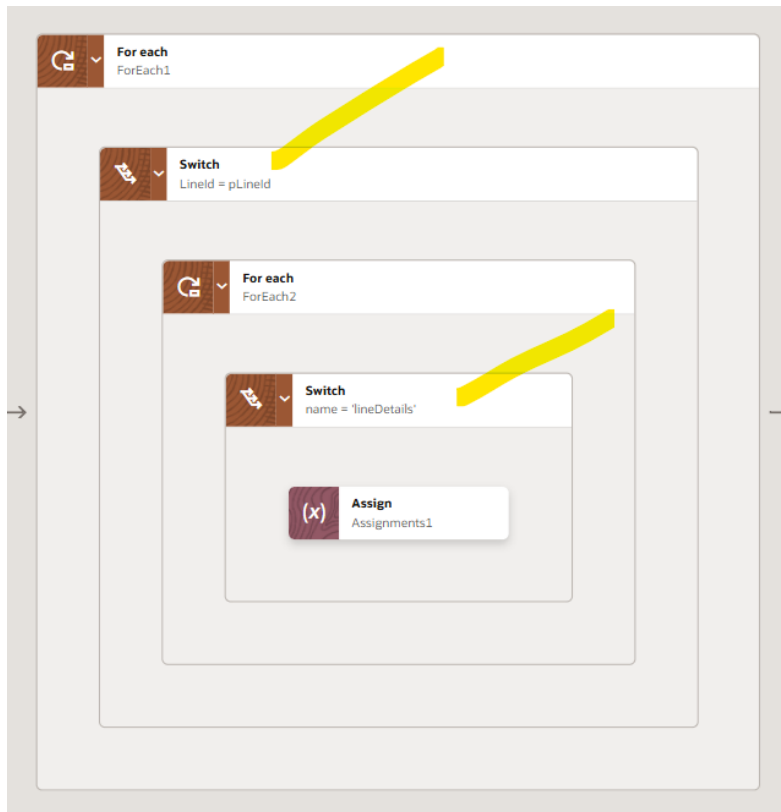
Close

Previous

Next



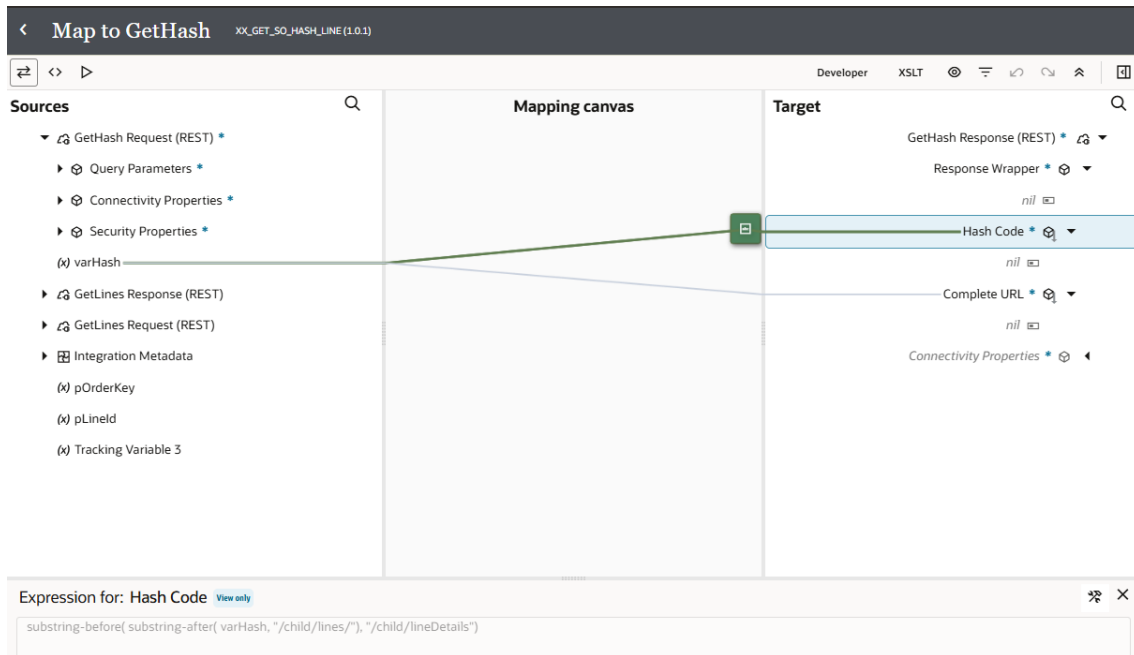
- b) We will iterate first in outer loop to find our line, then in inner loop to find our correct HREF node, using name.



c) Assignment.

This is a configuration window for an assignment. It features a top bar with a trash icon. Below it, there are three main sections: a 'Variable' field containing 'varHash', an 'Operation' dropdown menu set to 'Assign', and a 'Value' field containing the expression '\$f1_links/ns808:links/ns808:href'. To the right of the value field are icons for variable selection and a plus sign. A plus sign icon is also located at the bottom left of the window.

6) Response mapping.

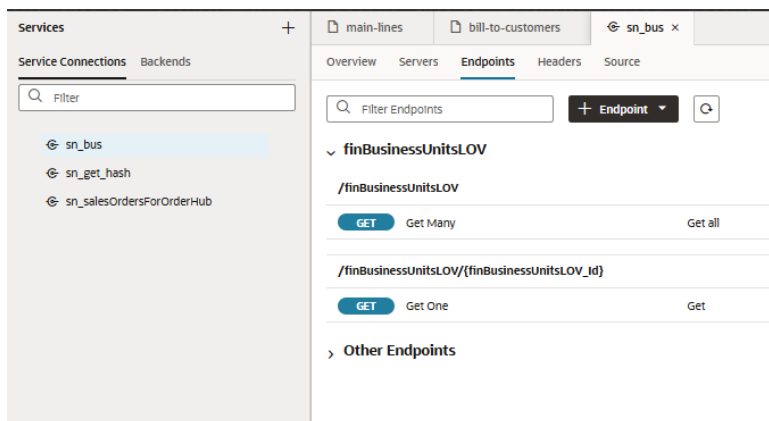


substring-before (substring-after (\$varHash, "/child/lines/"), "/child/lineDetails")

Preparatory steps Visual Builder.

Connections.

1) For Business Units



2) For Sales Orders.

Create Service Connection

Service Name ^{*}

sn_salesOrdersForOrderHub

https://eqiz.ds-fa.oraclepemos.com:443/fscmRestApi/rest/rv:384d7756-a949-40eb-a515-cbb5feb2b6a4/en/latest/9/describe Oracle Cloud Account

For better performance, select only the objects you want to use in your application.

salesOrdersForOrderHub x Select All 1 object(s) selected

- ☒ salesOrdersForOrderHub
- ☐ salesOrdersForOrderHubRequests

Getting Started | vcm_30 | 69 sn_salesOrdersForOrder... | x

Overview Servers Headers Source Endpoints Metadata

The endpoints are based on OpenAPI metadata, which we retrieved from the source URL dynamically. Tell me more

Filter Endpoints Edit Object Selection

salesOrdersForOrderHub

/salesOrdersForOrderHub			
GET	Get Many	Get all	getall_salesOrdersForOrderHub
POST	Create	Create	create_salesOrdersForOrderHub
/salesOrdersForOrderHub/action/applyHold			
POST	Create	applyHold	doall_applyHold_salesOrdersForOrderHub
/salesOrdersForOrderHub/action/releaseHold			
POST	Create	releaseHold	doall_releaseHold_salesOrdersForOrderHub
/salesOrdersForOrderHub/action/splitFulfillmentLine			
POST	Create	splitFulfillmentLine	doall_splitFulfillmentLine_salesOrdersForOrderHub
/salesOrdersForOrderHub/action/updateSchedulingAttribute			
POST	Create	updateSchedulingAttribute	doall_updateSchedulingAttribute_salesOrdersForOrderHub
/salesOrdersForOrderHub/action/updateActualDeliveryDate			
POST	Create	updateActualDeliveryDate	doall_updateActualDeliveryDate_salesOrdersForOrderHub
/salesOrdersForOrderHub/{salesOrdersForOrderHub_id}			

3) To OIC integration, to get the Hash Code.

First, our integration must be active.

Integrations Import Create

hash

Updated by juanje

2 integrations

Name	Version	Style	Last updated	Status
XX_GET_SO_HASH_LINE	1.0.1	Application	Jul 13, 2025 10:52:20 AM CEST	Active
XX_GET_SO_HASH_LINE	1.0.0	Application	Jul 12, 2025 07:54:27 PM CEST	Configured

Then backend should be as shown.

Services +

Service Connections Backends

Filter

Integration Applications

Oracle Cloud Applications

Service Connections + Service Connection

General

Name

ics

Backend Type

Integration Applications

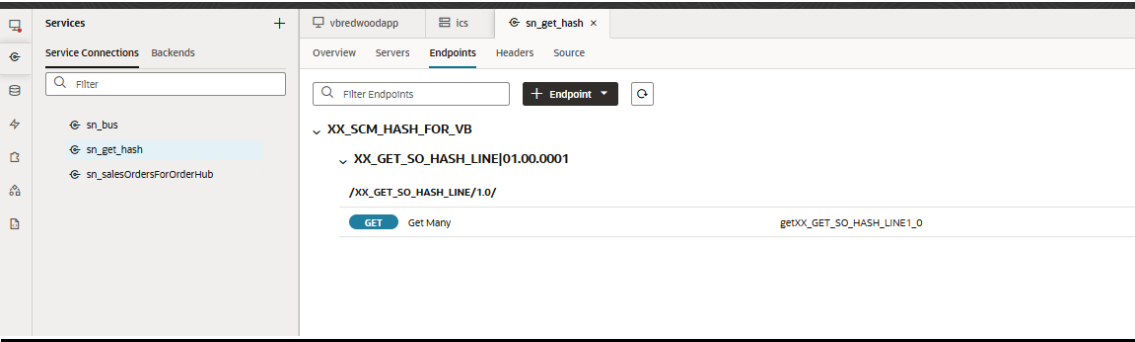
Description

Transforms [ⓘ] See examples

Source

None

This is the connection and Endpoint.



Steps in Visual Builder

We will build a basic master-detail-detail using “Welcome page templates” for the 3 levels for learning purposes.

First level. Sales order headers.

Redwood_cookbook_2026

Sales Orders Demo for Hash Codes

Redwood_cookbook_2026

Detail Lines

B.Unit

Q %

BusinessUnitName	RequestingBusinessUnitName	BuyingPartyName	Source Order System	Order	Source Order	TransactionType	BuyingPartyNumber	Status
France Business Unit	France Business Unit	Business World	ATG2	6216	6000314033		16080	Processing
France Business Unit	France Business Unit	Dixon Industries	LEG1	6220	6000314037		59300	Closed
France Business Unit	France Business Unit	Computer Service and Rentals	LEG1	6221	6000314039		10060	Closed
France Business Unit	France Business Unit	Dixon Industries	LEG1	6222	6000314040		59300	Closed
France Business Unit	France Business Unit	Dixon Industries	LEG1	6224	6000314042		59300	Closed
France Business Unit	France Business Unit	Dixon Industries	LEG1	6226	6000314044		59300	Processing
France Business Unit	France Business Unit	Business World	LEG1	6227	6000314045		16080	Partially Shipped
US1 Business Unit	US1 Business Unit	Business World	SEBL1	6232	2000314009		16080	Closed
US1 Business Unit	US1 Business Unit	Dixon Industries	SEBL1	6240	2000314010		59300	Closed

We will filter using Business Unit (Select single component) and Sales Order (Input text component).

B.Unit

Q %

BusinessUnitName	RequestingBusinessUnitName	BuyingPartyName	Source Order System	Order	Source Order
France Business Unit	France Business Unit	Business World	ATG2	6216	60003140
France Business Unit	France Business Unit	Dixon Industries	LEG1	6220	60003140
France Business Unit	France Business Unit	Computer Service and Rentals	LEG1	6221	60003140
France Business Unit	France Business Unit	Dixon Industries	LEG1	6222	60003140

```
1 <oj-sp-welcome-page id="oj-sp-welcome" background-color="{ $page.variables.backgroundColor }">
2   illustration-background="{ { require.toUrl($page.variables.illustrationBackground) } }"
3   illustration-foreground="{ { require.toUrl($page.variables.illustrationForeground) } }"
4   display-options.image-stretch="{ { $page.variables.imageStretch } }"
5   display-options.mobile-image="{ { $page.variables.mobileImage } }" display-options.responsive-padding="on"
6   page-title="Sales Orders Demo for Hash Codes" primary-action-label="Detail"
7   secondary-actions="{ { ['id':'Lines','label':'Lines','display':'Lines'] } }"
8   on-sp-primary-action="{ { $listeners.ojSpPrimaryAction } }"
9   on-sp-secondary-action="{ { $listeners.ojSpSecondaryAction } }" overline-text="Redwood_cookbook_2026"
10  description-text="Redwood_cookbook_2026"
11
12
13  <oj-select-single label-hint="B.Unit" class="oj-form-control-width-md"
14    data="{ { $variables.finBusinessUnitsOVListSDP } }" item-text="BusinessUnitName" value="{ { $variables.varBuild } }"></oj-select-single>
15
16  <oj-input-search value="{ { $variables.varSearch } }" class="oj-form-control-width-sm"></oj-input-search>
17
18  <oj-table style="height:550px; width: 100%" class="oj-table-stretch oj-flex-item" scroll-policy="loadMoreOnScroll"
19    data="{ { $variables.salesOrdersForOrderHubListSDP } }"
20    columns="{ { ['headerText':'BusinessUnitName','field':'BusinessUnitName'], ['headerText':'RequestingBusinessUnitName','field':'RequestingBusinessUnitName'] } }"
21    selection-mode="single" selection-required="true"
22    on-first-selected-row-changed="{ { $listeners.tableFirstSelectedRow } }"></oj-table>
23
24
25 </oj-sp-welcome-page>
26
```

We will access second level from here.

The screenshot shows the SAP Fiori Page Designer interface. The main canvas displays a 'Welcome Page Template' design. The design includes a header with the text 'Sales Orders Demo for Hash Codes' and a table with columns 'BusinessUnitName', 'RequestingBusinessUnitName', and 'BuyingPartyName'. The table contains data for 'France Business Unit' and 'US1 Business Unit'. The right sidebar shows the 'Properties' panel for the 'Welcome Page Template' with various configuration options like 'Display Options', 'Mobile Image', and 'Primary Action'.

The screenshot shows the SAP Fiori Page Designer interface with the 'Events' tab selected. The 'Events' tab lists two events: 'spPrimaryAction' and 'spSecondaryAction'. The 'spSecondaryAction' event is selected, and its 'Input Parameters' section shows a variable 'actionid' with the value '{{\$event.detail.actionid}}'. The 'Properties' panel on the right shows the 'Welcome Page Template' configuration.

The screenshot shows the SAP Fiori Page Designer interface with the 'Diagram' tab selected. The 'Diagram' tab shows a flow diagram with a 'Navigate To Page' action. The 'Navigate To Page' action is configured with parameters: 'pageid' (main-lines), 'porderkey' ({{\$variables.varOrderKey}}), and 'pordernumber' ({{\$variables.varOrderNumber}}). The 'Properties' panel on the right shows the 'Welcome Page Template' configuration.

Second level. Lines.

Lines

Sales Order Lines for:6216

Back

Bill To Customer

DisplayLineNumber	ProductNumber	OrderedUOM	Unit List Price	Ordered Quantity	Customer PO Line	InventoryOrganizationCode	InventoryOrganizationName	Item Type	ItemSut
1	AS89035	Ea	56166	1	Line # 1	000	Operations	4	Standan
2	AS89035	Ea	56166	1	Line # 1	000	Operations	4	Standan

Design mode.

Page Designer

main-start

main-lines

Live

Design

Code

main-lines

General

Events

Title

Description

Select Page Template

Preferred Layout

Grid

Flex

Block

Components

Business Objects

Services

Data

Filter

Welcome Page Template

Table

Lines

Sales Order Lines for:undefined

Description Text

Back

Bill To Customer

DisplayLineNumber	ProductNumber	OrderedUOM	Unit List Price	Ordered Quantity	Customer PO Line	InventoryOrg
1	AS89035	Ea	56166	1	Line # 1	000
2	AS89035	Ea	56166	1	Line # 1	000

We will calculate Hash Key as we navigate the records in this Action Chain.

Page Designer

main-start

main-lines

Live

Design

Code

Table

General

Data

Events

All

Quick Start

Components

Business Objects

Services

Data

Filter

Welcome Page Template

Table

Lines

Sales Order Lines for:undefined

Description Text

Back

Bill To Customer

DisplayLineNumber	ProductNumber	OrderedUOM	Unit List Price	Ordered Quantity	Customer PO Line	InventoryOrg
1	AS89035	Ea	56166	1	Line # 1	000
2	AS89035	Ea	56166	1	Line # 1	000

Table

first-selected-row = TableFirstSelectedRowChangeChain

Event

first-selected-row

Action Chain

TableFirstSelectedRowChangeChain

Input Parameters

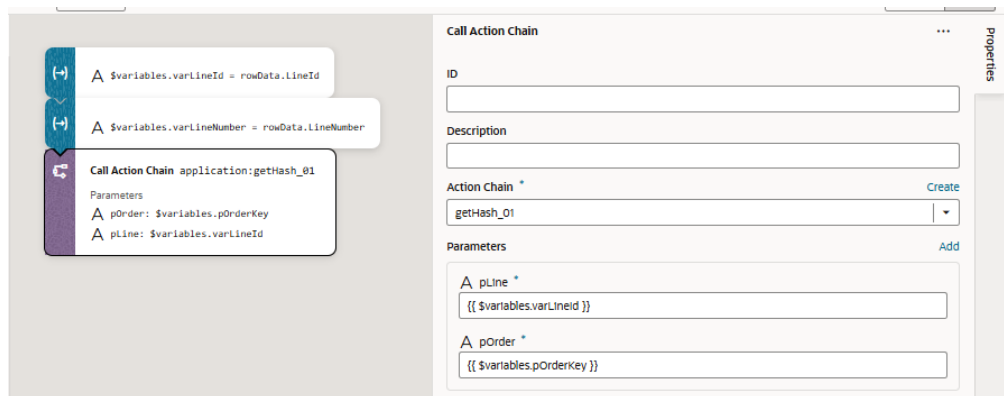
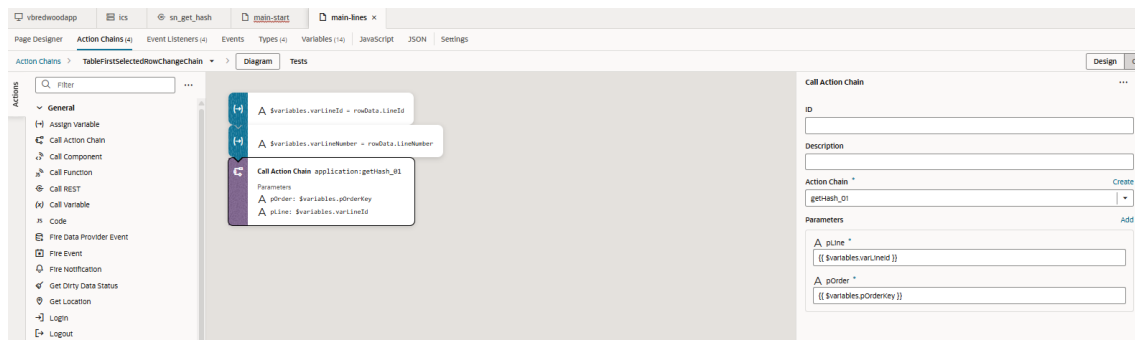
X: rowData *

[[event.detail.value.data]]

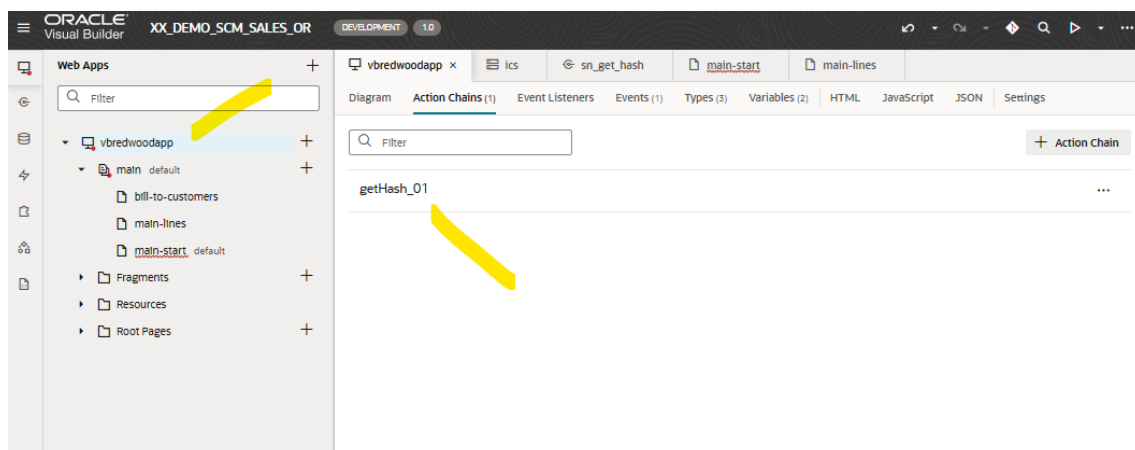
X: rowkey *

[[event.detail.value.key]]

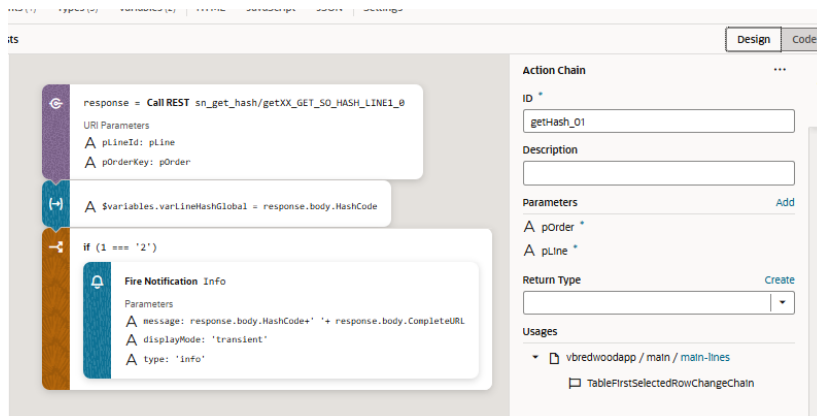
+ Event Listener



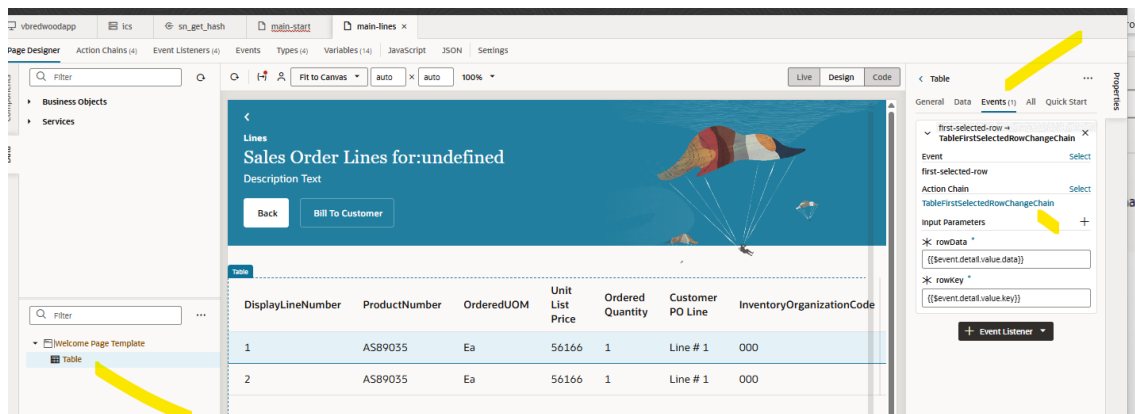
GetHash is at Application level, as shown.



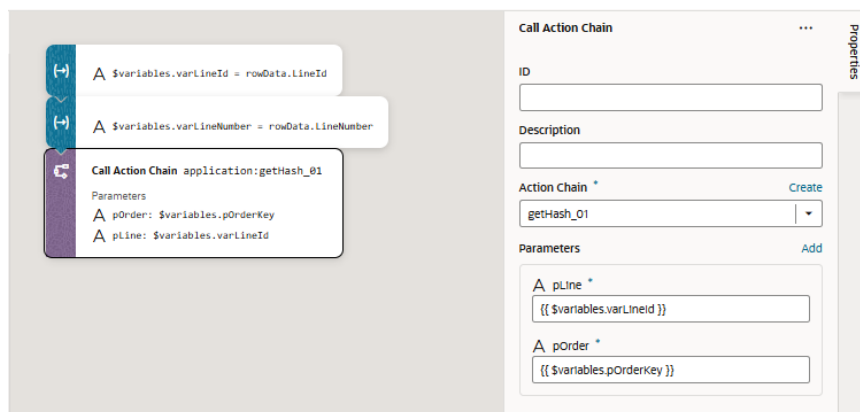
Here we call the OIC integration to find the Hask Key.



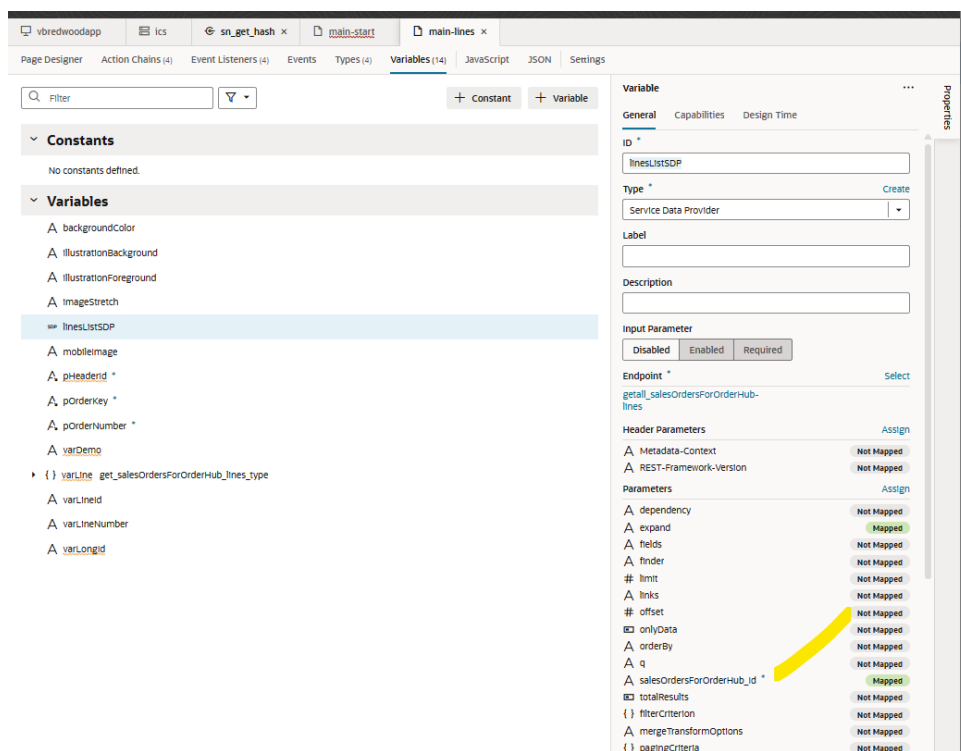
We will access second level page from here.



The Action Chain would show as follows.



The SDP would have this filter.



The detail would be this.

Map Variables To Parameters

Sources

Filter

Page (main-lines) +

Variables

background-color

A

illustrationBackground

A

illustrationForeground

A

imageStretch

A

linesListSDP

{ }

mobileImage

A

pHeaderId

A

pOrderKey

A

pOrderNumber

A

varDemo

A

varLine

{ }

varLineId

A

varLineNumber

A

varLongId

A

Target

Filter

Parameters

uriParameters

dependency

A

expand

A

fields

A

finder

A

limit

#

offset

#

onlyData

onlyData

orderBy

A

q

A

salesOrdersForOrderHubId*

A

totalResults

totalResults

filterCriteria

{ }

mergeTransformOptions

A

uriParameters.salesOrdersForOrderHub_id

A

1

\$variables.pOrderKey

Expression Static Content

Cancel

Save

Third level. Bill to data.

<

Bill to customers

Order.:ATG2:6000314033 Line..2

Back

AccountNumber	Address1	Address2	City	PostalCode	PartyName	Country	ContactName	State	ContactNumber
16080	Rue Saint Lazare		Paris	75004	Business World	FR	Helene Dubois		1041

18

Filter in SDP.

main-lines bill-to-customers x

Page Designer Action Chains (2) Event Listeners (2) Events Types (1) Variables (9) JavaScript JSON Settings

Filter

+ Constant + Variable

Constants

No constants defined.

Variables

- backgroundcolor
- billToCustomerListSDP
- illustrationBackground
- illustrationForeground
- imageStretch
- mobileImage
- pHash
- pLineNumber
- pOrder

Variable

General Capabilities Design Time

ID *

billToCustomerListSDP

Type *

Service Data Provider

Label

Description

Input Parameter

Disabled Enabled Required

Endpoint *

getall_salesOrdersForOrderHub-lines-billToCustomer

Header Parameters

- Metadata-Context Not Mapped
- REST-Framework-Version Not Mapped

Parameters

- Assign
- dependency Not Mapped
- expand Not Mapped
- fields Not Mapped
- # limit Not Mapped
- links Not Mapped
- # offset Not Mapped
- onlyData Not Mapped
- orderBy Not Mapped
- q Not Mapped
- salesOrdersForOrderHub_id* Mapped
- totalResults Not Mapped
- finder Not Mapped
- salesOrdersForOrderHub_lines_id* Mapped
- filterCriteria Not Mapped
- mergeTransformOptions Not Mapped
- pagingCriteria Not Mapped
- sortCriteria Not Mapped

Detail.

Map Variables To Parameters

Sources Target

Page (bill-to-customers) +

Variables

- backgroundcolor
- billToCustomerListSDP
- illustrationBackground
- illustrationForeground
- imageStretch
- mobileImage
- pHash
- pLineNumber
- pOrder

Parameters

- uriParameters
- dependency
- expand
- fields
- finder
- # limit
- links
- # offset
- onlyData
- orderBy
- q
- salesOrdersForOrderHub_id*
- salesOrdersForOrderHub_lines_id*
- totalResults
- filterCriteria

uriParameters.salesOrdersForOrderHub_lines_id

1 \$variables.pHash

Expression Static Content

Cancel Save

Conclusion

We have shown how to find and use Hash keys using OIC integration in our Redwood pages in Visual Builder.

We have used SCM Sales Orders for this exercise.

This technique would be needed in plenty of standard Fusion REST API's.

Technical

Code example	Comments
XX_DEMO_SCM_SALES_OR-1.0.zip	Visual builder app. used in this demo.
XX_GET_SO_HASH_LINE_01.00.0001.iar	Integration used to find the Hash Key