



Portal Configuration

Min Wu - Senior Technical Staff
September 2017

Metaphor for Syntax - a show or a play

Scenario: Portal configuration is considered as the description of a scenario on how users interact with the portal interface.

Stage: A scenario consists of multiple stages.

- One stage can lead to different stages.

Actor: A stage consists of multiple actors.

- Stage **layout** describes the relationship among actors in a stage.
- Actors interact with each other.
- Interaction may cause stage transfer.

Configuration files

Navigation / scenario JSON - main piece

Component / actor JSON - minimized version

Static content - HTML snippet

CSS

Component JSON

How to query data from a database table and display it in the interface?

- whereClause
- orderBy
- tableName
- sections **and** fields (**columns**)

Component fields - field type

Field type	Read only	Editable
string	<div>	<input type="text"/>
number, double, integer	<div>	<input type="number"/>
currency	Specified by locale setting	<input type="number"/>
percent	Ends with % and specified by locale settings	<input type="number"/>
date	Specified by locale settings	<input type="date"/>
time	Specified by locale settings	<input type="time"/>
datetime	Specified by locale settings	<input type="datetime-local"/>

Component fields - field type

Field type	Read only	Editable
textarea	HTML rendering	<textarea>
boolean	Font awesome icons: <input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>
radiobuttons	Font awesome icons: <input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/>
picklist	<div>	<select>
image	 url with communityPrefix	NA
link	<a>value url with linkTemplate and communityPrefix	NA

Reference field

Reference values can be displayed in two ways.

1. Reference value as regular field (such field should be ready only)
 - { "fieldName":
 "**genesis__Application__r.genesis__CL_Product__r.clcommon__Product_Name__c**",
 "fieldUi": {
 "fieldType": "**string**", "label": "Product Name", ...
 }, ... }

Reference field

Reference values can be displayed in two ways.

2. Reference value as an array of reference objects (such field should be editable as reference picklist)

- ```
{"fieldName": "Party_Type__c",
"relationType": "reference",
"relationName": "Party_Type__r",
"relationJson": actual JSON embedded,
"fieldUi": {
"fieldType": "reference", "label": "Party Type", ...}
```

Refer to another component JSON file to form the reference picklist

# Children relation

Rarely used because children relation data can be displayed as a separate actor

- "fieldName": **null**,  
"relationType": "**childrenRelation**",  
"relationName":  
**"clcommon\_DocCat\_Attachment\_Associations\_r"**,  
"relationJson": "**attachment\_list\_view**",  
"fieldUi": {"fieldType": **null**, "label": "Attachments", ...}

# Field validation

```
"fieldUi": {
 "required": true,
 "regex": "^(\w+)?(\d{9,14})$"
 "messages": {
 "requiredErrorText": "..."
 "regexErrorText": "..."
 ...
 }
}
```

# Field default value

```
"fieldName": "Party_Type__c",
"relationType": "reference",
"relationName": "Party_Type__r",
"relationJson": "portal2demo_party_type_reference",
"fieldUi": {
 "fieldType": "reference",
 "label": "Party Type",
 "defaultValue": "BORROWER",
```

# Reference field text

In an editable view, reference field loads its values from the backend in real time.

```
{
 "fieldName": "Party_Type__c",
 "relationType "relationName": "Party_Type__r",
 "relationJson": "portal2demo_party_type_reference",
 "fieldUi": {
 "fieldType "label": "Party Type",
 "defaultValue": "BORROWER",
 "messages": {
 "referenceLoadingText "referenceErrorText "requiredErrorText": "..."
 "regexErrorText": "..."
 }
 }
}
```

# Dropdown

Both reference field and picklist field can form a dropdown control.

Three attributes from `fieldUi` decide the dropdown behavior.

- `defaultValue`
- `prompt`
- `required`

# Dropdown default value

Only reference can have default value. If there is defaultValue, defaultValue will be displayed by default and there is no “empty” entry in the dropdown.

# Dropdown prompt

If there is no defaultValue and there is prompt, prompt will be displayed by default whose optionValue is “”.

- Therefore, if the dropdown is required and prompt is kept, when submit, the required error message will be displayed.

# Dropdown required

If there is no defaultValue nor prompt,

- If it is a picklist and if the field is required, the first option is displayed by default and there is no “empty” entry;
- Otherwise, an “empty” entry with optionValue “ ” will be displayed by default.

# New V3 field type

## enhancedPicklist

```
"fieldUi": {
 "fieldType": "enhancedPicklist",
 "label": "Actor Type",
 "autoFocus": true,
 "selectOptionSize": 15,
 "enhancedSelectOptions": [{
 "group": false,
 "selectOptions": [{...}, {...}, ...]
 }, {
 "group": true,
 "groupLabel": "Component",
 "selectOptions": [{...}, {...}, ...]
 }, {...}
}]
}
```

# fieldUi V3 attributes

“autoFocus”: true

For picklist, enhanced picklist, and different input field types

# Navigation JSON

Two main top levels: stages (scenario) and actors

```
{ stages: { ... }, # stage name/value mapping
 actors: { ... } } # actor name/value mapping
```

- stages:
- actors:
- settings:
- global:
- messages:

# Stage

Each stage **may have** layout **and** overlay.

- layout **is nested structure of** name (string) **and** children (array).
- overlay **is an array of layouts**, which specifies all modal dialogs that can be open in the current stage.

Use layout name **to refer to the corresponding actor** in the actors section.

- If no corresponding actor, the layout **is a <div>**.
- An actor **can contain other actors as children**, for example, a container tabs **contains multiple queue-view components**.

```
"WorkspaceHomePage": {
 "layout": {
 "name": "top",
 "children": [{
 "name": "AuthenticatedHeader"
 }, {
 "name": "LeftSidebar"
 }, {
 "name": "RightSidebar"
 }, {
 "name": "content",
 "children": [{
 "name": "ApplicationCardView"
 }]
 }]
 }
},
```

content static-content

component card-view

```
"ApplciationDetailPage": {
 "layout": {
 "name": "top",
 "children": [{
 "name": "AuthenticatedHeader"
 }, {
 "name": "content",
 "children": [{
 "name": "DocumentTabs",
 "children": [{
 "name": "queue-view"
 }],
 "name": "tabs"
 }],
 "name": "container"
 }],
 "name": "

"
 }]
},
"BusinessDocumentQueueView"
"PrincipalDocumentQueueView"
 "name": "preview",
 "children": [{
 "name": "PreviewPageInclude"
 }]
},
 "name": "page-include"
},
"AppcliationDetailPage": {
 "layout": {
 "name": "top",
 "children": [{
 "name": "AuthenticatedHeader"
 }, {
 "name": "content",
 "children": [{
 "name": "DocumentTabs",
 "children": [{
 "name": "queue-view"
 }],
 "name": "tabs"
 }],
 "name": "container"
 }],
 "name": "

"
 }]
},
"BusinessDocumentQueueView"
"PrincipalDocumentQueueView"


```

# Actor

Each actor has category and type

| category      | type                                                                  |
|---------------|-----------------------------------------------------------------------|
| component.tsx | list-view, queue-view, card-view, detail-view, base-panel, data-model |
| content.tsx   | static-content, page-include, authenticated-header                    |
| container.tsx | steps, tabs                                                           |

# component types

list-view: with column header (including sort indicator), one record per row

queue-view: no column header, with sort drop-down, one item per row

card-view: no column header, with sort drop-down, multiple items per row

detail-view: single record

data-model: for data reference only, no rendering

basic-panel: no backend data, may call remote custom API

# View Basic Single List View Page

Setup >> Visualforce Pages >> B >> BasicSingleListView >> Preview

Visualforce Page  
**basicSingleListView**

| Page Detail                             |                                       | Actions                                                  |                                                    | Properties                                       |                                                                                                        |
|-----------------------------------------|---------------------------------------|----------------------------------------------------------|----------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <input type="button" value="Edit"/>     | <input type="button" value="Delete"/> | <input type="button" value="Clone"/>                     | <input type="button" value="Where is this used?"/> | <input type="button" value="Show Dependencies"/> | <input style="border: 2px solid red; padding: 2px; margin-left: 10px;" type="button" value="Preview"/> |
| Label                                   | basicSingleListView                   | Name                                                     | basicSingleListView                                |                                                  |                                                                                                        |
| Namespace Prefix                        |                                       | Available for Salesforce mobile apps and Lightning Pages |                                                    | <input type="checkbox"/>                         |                                                                                                        |
| Require CSRF protection on GET requests | <input type="checkbox"/>              | Description                                              |                                                    |                                                  |                                                                                                        |
| Last Modified By                        | Min Wu, 8/14/2017 2:43 PM             | Created By                                               |                                                    | Min Wu, 8/8/2017 11:30 AM                        |                                                                                                        |

**Visualforce Markup** **Version Settings**

```
<apex:page showHeader="false" sidebar="false" standardStylesheets="false" applyHtmlTag="false" applyBodyTag="false" docType="html-5.0" c>
<apex:include pageName="clcommon__ResolveNamespace"/>
<html>
<head>
<link href="https://fonts.googleapis.com/css?family=Lato&subset=latin,latin-ext" rel="stylesheet"

```

# Developer Console to view Json

Developer Console >> basic\_single\_list\_view

Look for page ID copy / paste

Where name is

# component property

component has the following property

- structureName: component JSON
- whereClause: variable and value mapping, including all the variables used in the component JSON

```
"BorrowerCardView": {
 "category": "component",
 "type": "card-view",
 "property": {
 "structureName": "portal2demo_borrower_card_view",
 "whereClause": {
 "Guest_Account__c": "$global.attr(sessionId);"
 }
 }
}
```

# content types

static-content: HTML snippet

page-include: iframe

authenticated-header: a subclass of static-content, which can query the authenticated user information from the backend and store it as global values

# content property

content **has the following** property

- **contentName**: to retrieve the HTML snippet from the backend
- **url**: the source `src` of `<iframe>`

# container types

steps: wizard, indicating where the user is in a flow. But the steps are not clickable.

tabs: tabs are clickable to switch

No property

# appearance: a collection of key-value pairs of all look-and-feel related attributes

|              |                  |
|--------------|------------------|
| title        | sortableFields   |
| subTitle     | sortedField      |
| textBefore   | sortText         |
| textAfter    | groupBy          |
| blankMessage | showSectionTitle |

may include variable expressions and should be rendered as HTML

# appearance

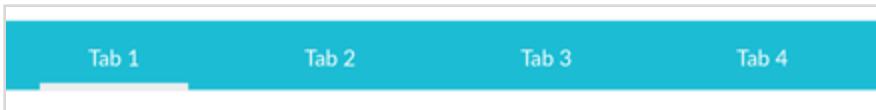
`sectionStyle: 'step' or 'layout'`

- **step: one section is visible at a time.**
- `layout`: all sections are visible.

`fieldStyle: 'stack' or 'side-by-side'`

- For each field, by default the label is about the value ('stack' style)
- 'side-by-side': label and value are side by side in a single row.

`tabStyle style: 'primary' or 'secondary'`



# Sorting - list-view, card-view, queue-view

```
"ApplicationCardView": {
 "appearance": {
 "columns": 3,
 "sortableFields": ["genesis_Maximum_Financed_Amount_c", "Term_Text_c",
 "genesis_Interest_Rate_c"],
 "sortedField": {
 "name": "genesis_Maximum_Financed_Amount_c",
 "ascending": false
 },
 "sortText": {
 "sortBy": "Sort By",
 "lowToHigh": "(Low to High)",
 "highToLow": "(High to Low)",
 "aToZ": "(A to Z)",
 "zToA": "(Z to A)"
 }
 }
}
```

Sort By  
Loan Amount (High to Low)

- Loan Amount (Low to High)
- ✓ Loan Amount (High to Low)
- Term (A to Z)**
- Term (Z to A)
- Rate (Low to High)
- Rate (High to Low)

| Product Name | Created Date | Loan Amount    | Term     | Rate   | APR    | Application Id     |
|--------------|--------------|----------------|----------|--------|--------|--------------------|
| Purchase     | 8/2/2017     | \$1,000,000.00 | 0 Months | 10.00% | 10.00% | a3m460000004K0uAAE |
| Purchase     | 8/4/2017     | \$1,000,000.00 | 0 Months | 10.00% | 10.00% | a3m460000004L0uAAE |

# groupBy - list-view, card-view, queue-view

Group the records by certain fields. Group title can be configured.

```
"BusinessDocumentQueueView": {
 "category": "component",
 "type": "queue-view",
 "property": {"structureName": "..."},
 "appearance": {
 "groupBy": {
 "field": "clccommon_Account_r.Name",
 "title": "
$this.attr(clcommon_Account_r.Name); |
$this.attr(clcommon_Account_r.Industry);
"
 },
 ...
 },
 ...
}
```

| Business Documents                    | Principal Doc |
|---------------------------------------|---------------|
| Acme Enterprise, Inc.   Manufacturing |               |
| Name                                  |               |
| Articles of Incorporation             |               |
| Industry                              |               |
| Manufacturing                         |               |
|                                       |               |
| Name                                  |               |
| Bank Statement                        |               |
| Industry                              |               |
| Manufacturing                         |               |

# groupBy with render - Groups with different titles

Rabo example: only show title for the closed-deal section.

```
"groupBy": {
 "field": "Is_Closed__c",
 "titles": [{
 "title": "<div><div>Gesloten
leningen</div><div>Hieronder vindt
u voorbeelden van leningen welke al
eerder via Rabo&Co zijn
gefinancierd. Voor deze leningen
is de inschrijfperiode verlopen. U
kunt zich niet meer inschrijven
voor deze leningen.</div></div>",
 "render":
 "$this.attr(Is_Closed__c); == true"
 }]
},
```

Totale financieri... € 300.000,00  
Rabobank deel (... € 200.000,00  
Open voor insc... € 100.000,00

Gesloten leningen  
Hieronder vindt u voorbeelden van leningen welke al eerder via Rabo&Co zijn gefinancierd. Voor deze leningen is de inschrijfperiode verlopen. U kunt zich niet meer inschrijven voor deze leningen.



Company Name Holding van Nijna...



Company Name Pukibi B.V.



Company Name Molkenboer Vastg...

# groupBy with render - Groups with different titles

## Another Rabo example

```
"NotificationQueueView": {
 "property": {
 "structureName": "notificationQueueViewV2"
 },
 "appearance": {
 "groupBy": {
 "field": "Required_c",
 "titles": [{
 "title": "Rabo&Co berichten",
 "render": "$this.attr(Required_c) == true"
 }, {
 "title": "Wilt u een melding ontvangen als...",
 "render": "$this.attr(Required_c) == false"
 }]
 }
}
```

# appearance

render: **string**

editable: **boolean**

- Whether it is an editable view or read-only view

expansionTop, expansionBottom, entryExpansionTop, entryExpansionBottom

- normal: 50px

messagePositions, entryMessagePositions

# appearance - Highlight

Highlight: true

- The list/queue/card view highlights the clicked record.

autoHighlight: true

- When a list/queue/card view is rendered, automatically highlight the first record.
- When `autoHighlight` is set, the `clickRecord` handler is called automatically.

```
"ContactQueueView": {
 "category": "component",
 "type": "queue-view",
 "property": {...},
 "appearance": {
 "highlight": true,
 "autoHighlight": "first"
 },
 "generalHandlers": {
 "clickRecord": {
 "assignments": [
 {
 "field": "$nav(ContactDetailView).property(whereClause.id);",
 "value": "$args[0].attr(id);"
 }
]
 }
 }
}
```

# appearance - Multiple selections

```
"AccountListView": {
 "category": "component",
 "type": "list-view",
 ...
 "appearance": {
 "columns": 4,
 "multipleSelections": true,
 "expansionTop": "normal",
 "sortableFields": ["Name", "Industry", "AnnualRevenue"],
 "sortedField": {"name": "Name", "ascending": true}
 },
 "componentActions": [{
 ...
 "enable": "$interaction.attr(selectedIds).count() > 0"
 }]
}
```

# appearance: message positions

messagePositions:

- loadingAnimation
- loadingError
- saveSubmitAnimation
- saveSubmitError
- saveSubmitSuccess
- uploadError

By default, all messages appear at the **top**.

entryMessagePositions

- saveSubmitAnimation
- saveSubmitError
- saveSubmitSuccess
- uploadError

# appearance

renderSections: sectionName → rendering **string**

renderFields: fieldName → rendering **string**

editableFields: fieldName → editable **string**

- **string** includes variable expressions and is evaluated as a boolean value. (examples in later slides)

# appearance

spanFields: ["Party\_Type\_\_c", "Street\_Address\_\_c"]

- In multi-column detail view or base panel, such fields span across columns with 100% width.

noLabelFields: ["clcommon\_\_Category\_Name\_\_c"]

- The field's label is not displayed.

collapsedHiddenFields: ["clcommon\_\_Description\_\_c", "clcommon\_\_Status\_\_c"]

- The fields will be hidden when an entry is collapsed.

collapsedHiddenActions: ["upload", "change"]

- The actions will be hidden when an entry is collapsed.

# appearance

columns: card-view, detail-view, basic-panel, **list-view**

authenticatedUserLabel: authenticated-header **only**

titles: container **only**

titleOrientation: container **only** - 'horizontal' **or** 'vertical'

current: container **only**

- current step, current tab or current open accordion section

# appearance

queue-view only (each queue entry can be collapsed.)

collapsible

collapsed

- collapsed: true by default queue entries are collapsed

collapsibleIcons: {}

- expanded
- collapsed

```
"BusinessDocumentQueueView": {
 "category": "component",
 "type": "queue-view",
 "property": {...},
 "appearance": {
 ...
 "collapsible": true,
 "collapsed": true,
 "collapsibleIcons": {
 "expanded": ">",
 "collapsed": ">"
 }
 },
},
```

appearance - “transient”: true

A component can be set “transient”: true so that its data will be cleared (reset) when

- The data has been successfully saved or submitted.
- The component is “switched away”.

Mainly used by edit forms with sensitive information, such as, login form, reset / set / change password form.

```
"LoginPage": {
 "category": "component",
 "type": "base-panel",
 "property": {"structureName": "..."},
 "appearance": {
 "transient": true,
 "editable": true,
 "title": "Log In"
 },
 "componentActions": [
 {
 "name": "login",
 "actionType": "submit",
 "label": "Log In",
 "remoteActionArguments": {
 "username": "$this.attr(username);",
 "password": "$this.attr(password);"
 },
 "remoteActionAPIname": "login",
 "successHandler": {
 "redirect": "$result;"
 }
 }
]
}
```

```
"ChangePasswordPanel": {
 "category": "component",
 "type": "base-panel",
 "property": {"structureName": "..."},
 "appearance": {
 "transient": true,
 "editable": true,
 "title": "Change Your Password"
 },
 "componentActions": [
 {
 "name": "save",
 "actionType": "submit",
 "label": "Change Password",
 "remoteActionArguments": {
 "oldPassword": "$this.attr(oldPassword);",
 "newPassword": "$this.attr(newPassword);",
 "verifyPassword": "$this.attr(verifyPassword);"
 },
 "remoteActionAPIname": "confirmPassword",
 "messages": {
 "showSubmitSuccess": true,
 "submitSuccessText": "..."
 }
 }
]
}
```

# authenticated-header

```
"AuthenticatedHeader": {
 "category": "content",
 "type": "authenticated-header",
 "property": {
 "contentName": "portal2demo_auth_header"
 },
 "appearance": {
 "authenticatedUserLabel": " $global.attr(userInfo).attr(firstName);
$global.attr(userInfo).attr(lastName);"
 }
}
```

# messages: text customization

loadingAnimationText

saveSuccessText

loadingFailureText

saveFailureText

savingAnimationText

submitSuccessText

submitAnimationText (remote API)

submitFailureText

submitExpirationText (polling only)

uploadFailureText

uploadFileTooLargeError

# messages: text customization (negative booleans)

By default, the animation and success messages are hidden unless users configure its `show-` to be `true`.

- `showLoadingAnimation`, `showSavingAnimation`,  
`showSubmitAnimation`, `showSavingSuccess`, `showSubmitSuccess`

By default, the error messages are displayed unless users configure its `hide-` to be `true`.

- `hideLoadingError`, `hideSavingError`, `hideSubmitError`

# Actions

Each **component** actor can have two types of actions

- componentActions: actions against the whole component, for example, save and cancel buttons for a detail view
- recordActions: actions against individual records in a component, for example, upload and preview button for each document category

Four actionTypes: save, submit (remote API), upload, normal

Each action may have

- render: string
- enable: string

```
"BorrowerDetailView": {
 "category": "component",
 "type": "detail-view",
 "componentActions": [{
 "name": "save",
 "actionType": "save",
 "label": "Save",
 "icon": "fa fa-floppy-o"
 }
],
 "recordActions": [
 {
 "name": "edit",
 "actionType": "edit",
 "label": "/",
 "icon": "fa fa-pencil"
 },
 {
 "name": "delete",
 "actionType": "delete",
 "label": "/",
 "icon": "fa fa-trash-o"
 }
]
},
"BusinessDocumentQueueView": {
 "category": "component",
 "type": "queue-view",
 "recordActions": [{
 "name": "preview",
 "actionType": "normal",
 "label": "/",
 "icon": "fa fa-expand"
 },
 {
 "name": "upload",
 "actionType": "upload",
 "label": "Upload",
 "icon": "fa fa-cloud-upload"
 }
]
}
```

# save action

name: unique name in the current component / actor

actionType

label,

style:

- "primary", "secondary", "normal", "warning", "iconic"

The image shows a login interface with a dark header bar. Below it, there are two input fields: one for 'E-mail Address' and one for 'Password'. At the bottom right is a teal-colored button labeled 'Log In'. Two pink arrows point from the text 'Text-style action' and 'Normal action' to the 'Log In' button. The 'Log In' button has a red border and contains white text.

Log In

E-mail Address \*

Password \*

Forgot Your Password? **Log In**

Text-style action      Normal action

# Action - position

top-left, top-right, bottom-left, bottom-right

# save action

remoteActionArguments: key/value pairs, before the save request is sent, set certain fields with certain values, and then save.

- Value can be variable expression, for example,

```
"remoteActionArguments": {"Guest_Account__c":
 "$global.attr(sessionId);"}
}
```

# save action

successHandler: what happens if save succeeds

failureHandler: what happens if save fails

- Handler configuration will be introduced later.

messages: specific messages for the current save action

- showSavingAnimation (boolean)
- savingAnimationText
- showSavingSuccess (boolean)
- hideSavingError (boolean)
- saveSuccessText
- saveFailureText

# submit action: call remote custom API

name (unique within actor), actionType, label, style

remoteActionArguments: key-value pairs submitted to remote custom API

remoteActionClassname: remote custom API name

**remoteActionAPIname: for built-in user-management APIs only**

remoteActionPrefix: remote custom API namespace

- "remoteActionArguments": {
  - "loanAmount": "\$state(OpportunityBidInputPanel).attr(loanAmount);",
  - "loanAppId": "\$state(OpportunityDetailView).attr(id);"}
- "remoteActionClassname": "PortalCreateInvestmentBooking"

# submit action

transaction

- If a remote action is transaction, a unique transaction Id is added to the `remoteActionArguments` when submitted.
- This attribute should be rarely used. Instead, users should depend on `remoteActionArguments` to encode the unique Id for each submission session.

## Polling configurations

- `pollingPeriod`: the time gap between two consecutive polls in milli-seconds
- `pollingMaxTries`

# Automatic action

**automatic**: the action will be automatically triggered.

- Both data-model and component whose render evaluated as false cannot have automatic action because action won't be rendered.

```
"PollingPanel": {
 "category": "component",
 "type": "base-panel",
 "componentActions": [
 {
 "name": "polling-status",
 "actionType": "submit",
 "automatic: true,
 "remoteActionArguments": {"applicationId": "$this.attr(applicationId)" },
 "remoteActionClassname": "PortalApplicationStatus1",
 "successHandler": {
 "assignments": [
 {
 "field": "$nav(OfferDetailView).property(whereClause.applicationId);",
 "value": "$result;"
 }
],
 "next": "OfferPage"
 },
 "failureHandler": {"next": "NoOfferPage"},
 "pollingPeriod": 3000,
 "pollingMaxTries": 3
 }
]
}
```

# submit action

successHandler

pendingHandler: what happens with PENDING response

failureHandler

messages: specific messages for the current submit action

- showSubmitAnimation (boolean)
- submitAnimationText
- showSubmitSuccess (boolean)
- hideSubmitError (boolean)
- submitSuccessText
- submitFailureText
- submitExpirationText

# submit action - user management

remoteActionAPIname

## 5 user management APIs

- Login (username, password)
- Logout ()
- resetPassword (email or username): In the forgot-password page, users provide email or username so that Salesforce sends them an email with the set-password link.
- confirmPassword (oldPassword, newPassword, verifyPassword): change password
- setPassword (newPassword, verifyPassword): new users set their password or existing users set a new password because they forget their current one.

# upload action: only available to authenticated users

name (unique within actor), actionType, label, style

uploadActionArguments: one built-in key-value pair

- attachmentParentId

REST call to upload files uses the sessionId from authenticated userInfo

# upload action

Specify remote action to deal with the returned attachment Id(s)

- two built-in variables:
- \$uploadedAttachmentIds ; - if the remote action needs attachment Id array
- \$uploadedAttachmentId; - if the remote action needs single attachment Id

remoteActionArguments

remoteActionClassname

remoteActionPrefix

```
{
 "name": "upload",
 "actionType": "upload",
 "label": "Upload",
 "uploadActionArguments": {
 "attachmentParentId": "$this.attr(Attachment_Parent_Id_c);"
 },
 "remoteActionArguments": {
 "documentCategoryId": "$this.attr(id);",
 "attachmentIds": "$uploadedAttachmentIds;"
 },
 "remoteActionClassname": "PortalLinkAttachmentList",
 ...
}
```

# upload action

successHandler: if upload action or remote action succeeds

pendingHandler: if remote action returns PENDING status

failureHandler: if upload action or remote action fails

messages: specific messages for the current upload action

- uploadFailureText
- uploadFileTooLargeError
- showSubmitAnimation (boolean)
- submitAnimationText
- showSubmitSuccess (boolean)
- hideSubmitError (boolean)
- submitSuccessText
- submitFailureText
- submitExpirationText

```
"AttachmentListView": {
 "category": "component",
 "type": "list-view",
 "property": {
 "structureName":
 "portal2demo_attachment_list_view"
 },
 "componentActions": [{
 "name": "upload",
 "actionType": "upload",
 "label": "Upload",
 "uploadActionArguments": {
 "attachmentParentId":
 "0014600000VEkGb"
 },
 "successHandler": {
 "updates": [{
 "name": "$nav.reloads.add;",
 "value": "AttachmentListView"
 }]
 }]
 }]
}
```

Success handler  
of upload action

```
"recordActions": [
 {
 "name": "upload",
 "actionType": "upload",
 "label": "Upload",
 "uploadActionArguments": {
 "attachmentParentId":
 "$this.attr(Attachment_Parent_Id__c);"
 },
 "remoteActionArguments": {
 "documentCategoryId": "$this.attr(id);",
 "attachmentIds": "$uploadedAttachmentIds;"
 },
 "remoteActionClassname": "PortalLinkAttachmentList1",
 "successHandler": {
 "assignments": [{
 "field":
 "$this.attr(clcommon_DocCat_Attachment_Associations_r)
 ;",
 "value": "$result;"
 }]
 },
 "messages": {
 "uploadFileTooLargeError": "Cannot upload this file.
The size is too large (> 25Mb).",
 "uploadFailureText": "Fail to upload"
 }
 }]
```

Success handler  
of remote action

## normal action

name, actionType, label

handler: what happens if this action is triggered

# Action appearance - V3

In Portal V3, the following attributes are grouped under appearance

```
"appearance": {
 "label": "...",
 "style": "...",
 "position": "...",
 "tooltip": "...",
 "render": "...",
 "enable": "..."
}
```

# Disclosure

Disclosures can be associated with component actors and their actions.

Each component actor can specify its associated disclosures as an array of  
`{name, render, enable}`

- When a disclosure is hidden, it is automatically unchecked.

`"Disclosures": [ { "name": "FinancialDisclosure" } ]`

# Disclosure

Each action in a component actor can be associated with the component's disclosures

disclosureEnable: an action is enabled only when certain disclosures are checked.

- disclosureEnable should be treated separately from enable because disclosure involves user's commitment.
- disclosureEnable specification only involves the disclosures in the current component, \$disclosure(disclosureName)

disclosureNames: an array of disclosure names sent along with save, submit, or upload action

```
"FinancialDetailView": {
 ...
 "disclosures": [{
 "name": "FinancialDisclosure"
 }],
 "componentActions": [{
 "name": "save",
 "actionType": "save",
 "label": "Save",
 "disclosureEnable": "$disclosure(FinancialDisclosure);",
 "disclosureNames": ["FinancialDisclosure"],
 ...
 }]
```

# Handler

Specifies what happens when

- a normal action is triggered, or
- response or failure is received after a save, submit or upload action is triggered.

Each handler has four parts

- assignments: assign given values to certain fields
- updates: update certain fields using given values through update methods
- **next: next stage**
- redirect: redirect outside to a different URL, for example, after a user is successfully authenticated at the login page.

# Handler assignments

An array of { "field":..., "value": ... }

field: a variable expression

- An actor's property or appearance
  - `$nav(actorName).property(fieldName);`
  - `$nav(actorName).appearance(fieldName);`
- The current actor's data
  - `$this.attr(fieldName);`
- Other actor's data
  - `$state(actorName).attr(fieldName);`
- Global data
  - `$global.attr(fieldName);`

# Handler assignments

value: a JS value or a variable expression

- The arguments from built-in handlers
  - `$args[index];`
  - `$args[index].attr(fieldName);`
- The remote response
  - `$result;`
  - `$result.attr(fieldName);`

# Handler assignments

value: a JS value or a variable expression

- An actor's property or appearance
  - `$nav(actorName).property(fieldName);`
  - `$nav(actorName).appearance(fieldName);`
- The current actor's data
  - `$this.attr(fieldName);`
- Other actor's data
  - `$state(actorName).attr(fieldName);`
- Global data
  - `$global.attr(fieldName);`

# Handler assignments

value can be a string with variable-expression interpolation.

```
"assignments": [{
 "field": "$nav(PreviewPageInclude).property(url);",
 "value":
 "/servlet/servlet.FileDownload?file=$this.attr(clcommon__DocCat_Attachment_Associations_r).attr(clcommon__Attachment_Id_c);"
}]
```

\* set the url of the document preview iframe using the attachment Id from the first record of the DocCat\_Attachment children relationship

```
"assignments": [{
 "field": "$nav(FinancialDetailView) .property(whereClause.recordId) ;",
 "value": "$result.attr(id) ;"
}, {
 "field": "$nav(ApplicationSteps) .appearance(current) ;",
 "value": 1
}]
```

\* After save action succeeds, the saved record is returned as response. Use the Id of the saved record to query property (whereClause.recordId) the data in FinancialDetailView.

\* Move to the second step of ApplicationSteps.

```
"assignments": [{
 "field": "$this.attr(clcommon__DocCat_Attachment_Associations__r) ;",
 "value": "$result;"
}]
```

\* The remote API returns an array of all the children relationship records from DocCat\_Attachment\_Association. Use the response to set the component's children relationship data.

```
"assignments": [{
 "field": "$nav(BorrowerCardView).property(whereClause.Guest_Account__c);",
 "value": "$global.attr(sessionId);"
}, {
 "field": "$nav(GuarantorCardView).property(whereClause.Guest_Account__c);",
 "value": "$global.attr(sessionId);"
}, {
 "field": "$nav(ApplicationSteps).appearance(current);",
 "value": 2
}]
```

\* sessionId is a global attribute specified in navigation JSON. Use it to query BorrowerCardView and GuarantorCardView.

\* Move to the third step of ApplicationSteps.

```
"assignments": [{
 "field": "$nav(BorrowerDetailView) .property(whereClause.recordId) ;",
 "value": "$this.attr(id) ;"
}, {
 "field": "$nav(ApplicationSteps).appearance(current) ;",
 "value": 0
}]
```

\* Use the `id` of the first record in the current component to query `property(whereClause.recordId)` the data in `BorrowerDetailView`.

```
"assignments": [{
 "field": "$nav(BorrowerDetailView) .property(whereClause) ;",
 "value": null
}]
```

\* Set `BorrowerDetailView's` `whereClause` to `null` so that the component displays a new view.

```
"clickRecord": {
 "assignments": [{
 "field": "$nav(BusinessDocumentQueueView) .property(whereClause.applicationId) ;",
 "value": "$args[0].attr(genesis_Application_c);"
 }],
 "trigger": "APPLICATION_DETAILS"
}
```

\* clickRecord is a built-in handler in component actor. It takes the record being clicked as the first and only argument. Use the Application\_c value in the first argument (clicked record) to set the query.

```
"componentActions": [{
 "name": "save",
 "actionType": "submit",
 "remoteActionClassname": "sampleRemoteApi",
 "successHandler": {
 "assignments": [{
 "field": "$nav(ApplicationSteps).appearance(current);",
 "value": 1
 }, {
 "field": "$state(FinancialDetailView);",
 "value": "$result;"
 }]
 }
}]
```

\* Use the response from a remote API (\$result;) to set data of a component.

# Handler updates

Sometimes, we need to update values based on what they are. Then we cannot use assignments.

Updates are for special cases ONLY. (We have three built-in updates so far.)

- Add reloads: `$nav.reloads.add` with an `actorName` or an array of `actorNames`
- Add overlay (modal dialogs): `$nav.overlay.add` with a layout name or an array of layout names
- **Remove overlay (modal dialogs): `$nav.overlay.remove` with a layout name or an array of layout names** (need it because some actions (not the built-in close button) in overlay can cause it to close.)

An array of { "name": ..., "value": ... }

```
"successHandler": {
 "assignments": [...],
 "updates": [{{
 "name": "$nav.reloads.add;",
 "value": ["BorrowerCardView", "GuarantorCardView"]
 }}]
}
```

\* After save succeeds, mark `reload` to two actors `BorrowerCardView` and `GuarantorCardView`. When we display those actors again, they will query the data from the server (even though they have data already since the data is considered stale) and remove their own `reload` flag.

```
"ApplicaitonDetailPage": {
 "layout": {...},
 "overlay": [{
 "name": "preview",
 "children": [{
 "name": "PreviewPageInclude"
 }]
 }]
}

"recordActions": [{
 "name": "preview",
 "actionType": "normal",
 "handler": {
 "assignments": [...],
 "updates": [{
 "name": "$nav.overlay.add",
 "value": "preview"
 }]
 }...
}
```

\* `overlay` is an array of layouts, which specifies all modal dialogs that can be open in the current stage.

\* When the preview action is triggered, the preview layout in `overlay` is added to the current overlay array, and thus displayed as a modal dialog.

By default, the overlay array is empty, and `$nav.overlay.add`, thus no modal dialog is displayed.

# Handler next and redirect

Can be variable expression too

```
"componentActions": [{
 "name": "login",
 "actionType": "submit",
 "label": "Log In",
 "remoteActionArguments": {
 "username": "$this.attr(username);",
 "password": "$this.attr(password);"
 },
 "remoteActionAPIname": "login",
 "successHandler": {
 "redirect": "$result;"
 }
}]

"redirect": url

"redirect": {
 "url": url,
 "target":
 "_blank"
}
```

# generalHandlers

Actor may have `generalHandlers`, which can be triggered without actor's actions (`componentActions` and `recordActions`).

- `navigate`: Actor with HTML snippet can trigger `navigate` action, which needs to be handled by `navigate generalHandler`, which is a map of `navigate name` and the corresponding handler.
  - `<tag navigate="navigateName"/>`
  - `"actorName": {"generalHandlers": {"navigate": {"navigateName": { ... }}}`

# generalHandlers

- clickRecord: if clickRecord is specified under generalHandler, clicking an entry in a component actor will trigger the handler. clickRecord takes the clicked entry data as its only argument.

```
"ApplicationCardView": {
 "category": "component",
 "type "property": {...},
 "generalHandlers "clickRecord": {
 "assignments": [{
 "field "value }],
 "next":
 }...
 }
 "ApplicationDetailView"
}
```

# generalHandlers - navigate

```
"SimpleLayout_HeaderContentZone": {
 "category": "content",
 "type": "static-content",
 "property": {
 "contentName": "portal2demo_public_header"
 },
 "generalHandlers": {
 "navigate": {
 "loginPage": {
 "next": "LoginPage"
 },
 "homePage": {
 "next": "ApplicationPage"
 }
 }
 }
}
}

<div class="links">
 <div class="nav-link" navigate="homePage" <div class="nav-link" navigate="loginPage"</div>
```



# pendingHandler is special

If an action has pendingHandler, the main purpose is to use the \$result from the PENDING response to update the remoteActionArguments and re-send the same request.

A new variable name  
is introduced \$remoteArgs

```
{ "actionType": "submit",
 "remoteActionArguments": {
 "guestAccountId": "$global.attr(sessionId);",
 "applicationId": "$this.attr(applicationId);"
 },
 "pollingPeriodpollingMaxTries": 3,
 "pendingHandler": {
 "assignments": [
 {"field": "$remoteArgs(applicationId);",
 "value": "$result;"}
]
 }
}
```

# render

render: string **can be assigned to**

- component, content, container (`render under` appearance)
- section (`renderSections under` appearance)
- field (`renderFields under` appearance)
- action (`render attribute`)
- disclosure (`render attribute`)

# render

render string can be evaluated against

- The data from current actor
  - Field value: \$this.attr(fieldName);
  - Aggregated value: \$this.count();, \$this.sum(fieldName);
  - Concatenation: \$this.attr(fieldName).count();
- The data from all actors:
  - Field value: \$state(actorName).attr(fieldName);
  - Aggregated value: \$state(actorName).count();,  
\$state(actorName).sum(fieldName);
  - Concatenation: \$state(actorName).attr(fieldName).count();
- The global data:
  - Field value: \$global.attr(fieldName);

# render

render string is based on [math.js syntax](#).

```
"renderSections": {
 "borrower": "$this.attr(Party_Type__r).attr(Name) ; == \"BORROWER\"",
 "guarantor": "$this.attr(Party_Type__r).attr(Name) ; == \"GUARANTOR\""
}

"renderFields": {
 "Guest_Account__c": "false"
}

"render": "$this.attr(Company_Name__c) ; != undefined"
```

```
"renderSections": {
 "common": "false",
 "corporation": "$this.attr(Legal_Entity_Type__r.Name); ==
\\\"Corporation\\\" or $this.attr(Legal_Entity_Type__r.Name); == \\\"Limited Liability
Company\\\" or $this.attr(Legal_Entity_Type__r.Name); == \\\"Partnership\\\" or
$this.attr(Legal_Entity_Type__r.Name); == \\\"Trust\\\" or
$this.attr(Legal_Entity_Type__r.Name); == \\\"Government\\\" or
$this.attr(Legal_Entity_Type__r.Name); == \\\"Association\\\",
 "individual": "$this.attr(Legal_Entity_Type__r.Name); == \\\"Sole
Proprietorship\\\" or $this.attr(Legal_Entity_Type__r.Name); == \\\"Individual\\\""
}

"render": "$this.attr(clcommon__DocCat_Attachment_Associations__r).count(); >
0",
```

# enable

enable: string **can be assigned to**

- field (enableFields **under** appearance)
- action (enable **attribute**)
- disclosure (enable **attribute**)

# Action specific messages

Display the server-side error message in the component:

```
"messages": {
 "submitFailureText": "$serverMessage;"
}
```

# Content appearance

messagePositions:

- loadingAnimation
- loadingError

By default, all messages appear at the **top**.

# String interpolation

The following types of visible text support string interpolation and are rendered as HTML:

- component:
  - title, subTitle, textBefore, textAfter, blankMessage
  - section title
  - field label
- content: **html content**
- container: titles
- action label
- messages

# String interpolation

String interpolation can specify the converted value type as the second argument.

```
"title": "Total amount:
$this.sum(loan__Remaining_Investment_Amount__c, currency);"
```

# String interpolation

The following HTML attributes support string interpolation:

- Section names in component JSON (used by CSS selectors)
  - "sections": [ {"name": "\$this.attr('clcommon\_Status\_c')"}, ...]
- linkTemplate in component JSON
  - "fieldUi": {"linkTemplate": "/servlet/servlet.FileDownload?file=\$this.attr('id')"}, ...}
- This list can grow.

# Variable expression

Variable expression **starts with \$ and ends with ;**.

- \$global
- \$nav (**navigation update**: property, appearance, reload, overlay)
- \$state (**data access**)
- \$this (**data access**)
- \$result (**remote response data**)
- \$args (**argument list to special methods, e.g., clickRecord**)
  - For `clickRecord`, the first argument is the record that a user has clicked.
- \$interaction
- \$disclosure
- \$remoteArgs

# Special variables

`$globalSessionId;`

- A unique GUID for the current session
- Usually set as a global value in navigation JSON
- "global": { "sessionId": "\$globalSessionId;" }

`$uploadedAttachmentIds;` an array of attachment Ids returned by upload

`$uploadedAttachmentId;` an attachment Id returned by upload

# settings: top-level attribute in navigation JSON

```
"settings": {
 "locale": "en-us",
 "currency": "USD",
 "minimumFractionDigits": 2,
 "maximumFractionDigits": 2,
 "leaveApplicationWarning": true
}
```

- locale used to display date, time, and datetime value
- MinimumFractionDigits, maximumFractionDigits used to display percent value
- currency, locale, and minimumFractionDigits, maximumFractionDigits used to display currency value

# global: top-level attribute in navigation JSON

```
"global": {"sessionId": "$globalSessionId;"}
```

userInfo is added to global by authenticated-header, so that you can use userInfo in navigation JSON

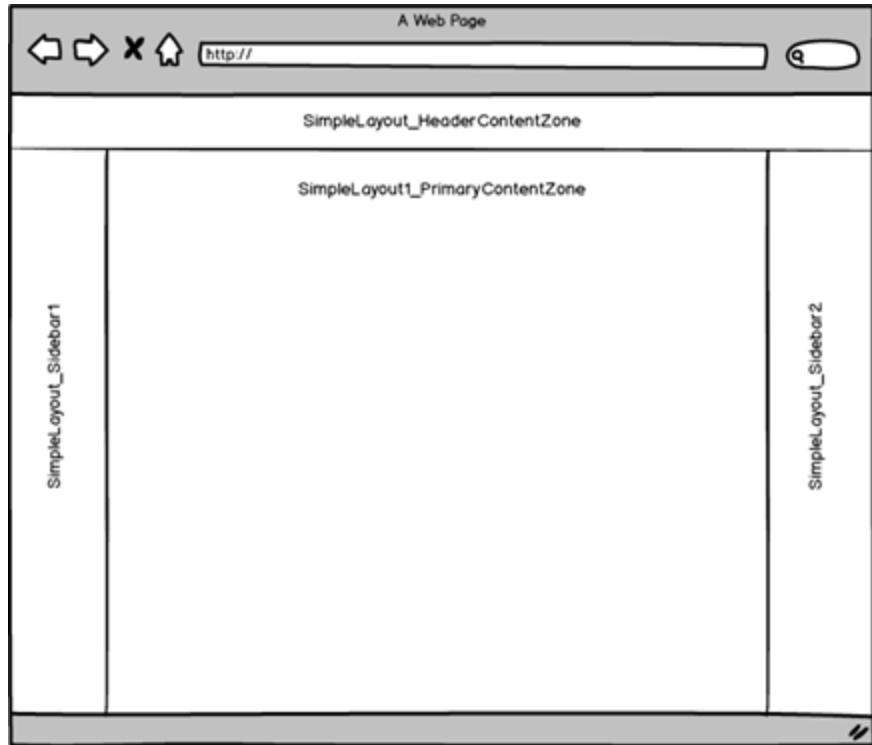
- \$global.attr(userInfo).attr(sessionId);
  - Note that this sessionId is different from the sessionId in the first line, which can be used before user logs in without userInfo.
- \$global.attr(userInfo).attr(firstName);
- \$global.attr(userInfo).attr(lastName);
- etc.

# messages: global message configuration

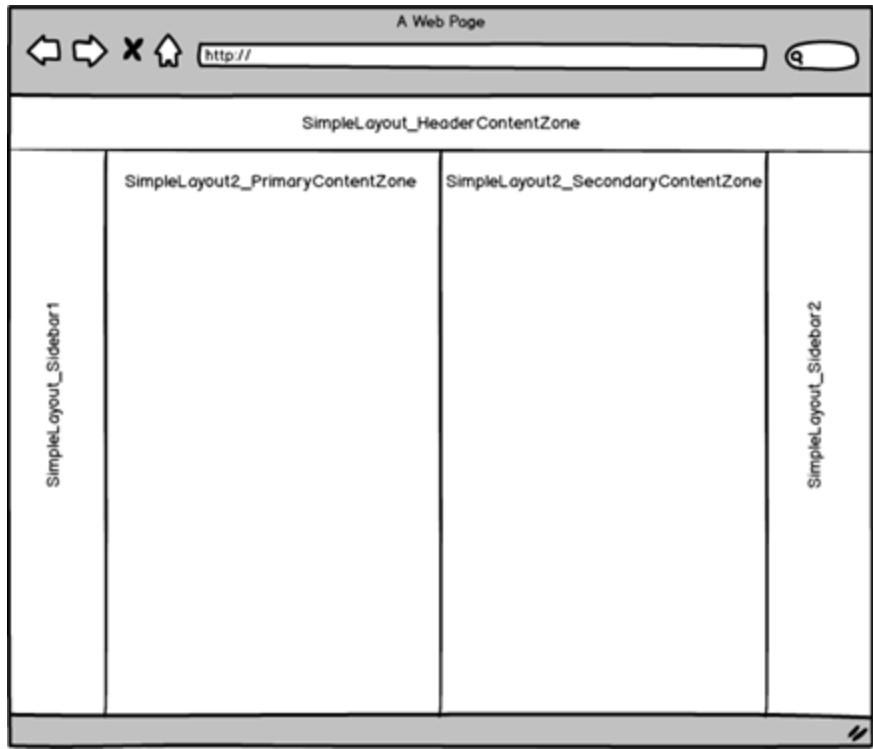
- loadingAnimationText
- savingAnimationText
- submitAnimationText
- loadingFailureText
- saveSuccessText
- saveFailureText
- submitSuccessText
- submitFailureText
- submitExpirationText
- uploadFailureText
- uploadFileTooLargeError
- leaveApplicationWarning (only can be displayed in MS IE and Edge)

# Layout Templates

SimpleLayout



```
"stageName": {
 "layout": {
 "name": "SimpleLayout_Top",
 "children": [{
 "name": "SimpleLayout_HeaderContentZone"
 }, {
 "name": "SimpleLayout_Sidebar1"
 }, {
 "name": "SimpleLayout_Sidebar2"
 }, {
 "name": "SimpleLayout1_PrimaryContentZone",
 "children": [{...}]
 }]
 }
}
```



```
"stageName": {
 "layout": {
 "name": "SimpleLayout_Top",
 "children": [
 {
 "name": "SimpleLayout_HeaderContentZone"
 },
 {
 "name": "SimpleLayout_Sidebar1"
 },
 {
 "name": "SimpleLayout_Sidebar2"
 },
 {
 "name": "SimpleLayout2_PrimaryContentZone",
 "children": [...]
 },
 {
 "name": "SimpleLayout2_SecondaryContentZone",
 "children": [...]
 }
]
 }
}
```

## EXPLORER

## OPEN EDITORS

- webpack.config.js
- widget-decorator.tsx src/framework...
- component.tsx src/framework...
- (1) basic\_list\_edit\_view.json src/a...
- (1) detail\_view\_with\_related\_item...
- (1) basic\_single\_list\_view.json sr...
- (1) investorV2.json src/applicatio...
- P layout.scss src/applications/r...
- P constants.scss src/application...

## PORTAL2

- (1) flow\_loan\_detail\_panel.js...
- navigation
  - (1) basic\_list\_edit\_new\_view....
  - (1) basic\_list\_edit\_view.json
  - (1) basic\_queue\_detail\_view....
  - (1) basic\_single\_card\_view.js...
  - (1) basic\_single\_list\_view.json
  - (1) detail\_view\_with\_related\_...
  - (1) flow\_consumer\_steps.json
  - (1) message\_save.json
  - (1) sortable\_card\_edit\_view.j...
  - (1) sortable\_single\_card\_vie...
  - (1) sortable\_single\_list\_view...
- framework
- actions

```
1 {
2 "start": "Homepage",
3 "stages": {
4 "Homepage": {
5 "layout": {
6 "name": "SimpleLayout_Top",
7 "children": [
8 {
9 "name": "SimpleLayout_HeaderContentZone"
10 },
11 {
12 "name": "SimpleLayout_Sidebar1"
13 },
14 {
15 "name": "SimpleLayout_Sidebar2"
16 },
17 {
18 "name": "SimpleLayout1_PrimaryContentZone",
19 "children": [
20 {
21 "name": "AccountListView"
22 }
23]
24 }
25],
26 }
27 },
28 "actors": {
29 "AccountListView": {
30 "category": "component",
31 "type": "list-view",
32 "property": {
33 "structureName": "basic_accounts"
34 },
35 "appearance": {
36 "columns": 4
37 }
38 }
39 }
40 }
41 }
```

```
navigation JSON
{
 "start": ...
 "stages": {"stageName": {...}}
 "actors": {"actorName": {...}}
 "settings":
 "global":
 "messages":
}

{
 "layout": stage
 "overlay": [layout]
}

 {
 "actorName": {
 "category":
 "type":
 "property":
 "appearance":
 "messages":
 "componentActions": [...]
 "recordActions": [...]
 "generalHandlers": {...}
 "disclosures": {...}
 }
 }

 {
 "navigate": { "navigateName": {...} }
 "clickRecord": {...}
 }

 {
 "assignments": [...]
 "updates": [...]
 "next": next_stage_name
 "redirect": url
 }

 {
 "name":
 "actionType":
 "label":
 "style":
 "render":
 "enable":
 "remoteActionArguments":
 "remoteActionClassname":
 "remoteActionPrefix":
 "remoteActionAPIname":
 "uploadActionArguments":
 "successHandler": {...}
 "failureHandler": {...}
 "pendingHandler": {...}
 "handler": {...}
 "disclosureNames":
 "disclosureEnable":
 "messages":
 "pollingPeriod":
 "pollingMaxTries":
 }
}
```

# User management

# Advanced feature - dialogs

Disclosure dialog

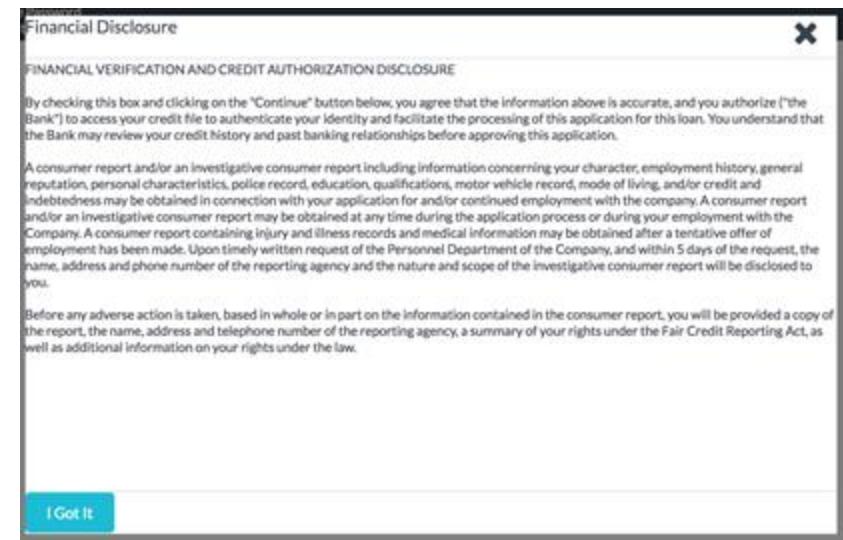
```
export interface IDialog {
 type: string;
 title?: string;
 content?: string;
 showCloseButton: boolean;
 closeText?: string;
 showConfirmButtons: boolean;
 confirmText?: string;
 cancelText?: string;
}
```

Action confirmation dialog

# Disclosure dialog

In order to support mobile UI, disclosure can be displayed in a modal dialog

Borrower Information	Financial Information	Additional Borrowers	Summary
Total Sales/Revenue	Total Assets		
Total Debt	Total Liabilities		
Net Worth of Business	Cash		
<input type="checkbox"/> I acknowledge and agree to the <a href="#">Financial Disclosure</a> .			
<a href="#">Skip</a>		<a href="#">Save</a>	



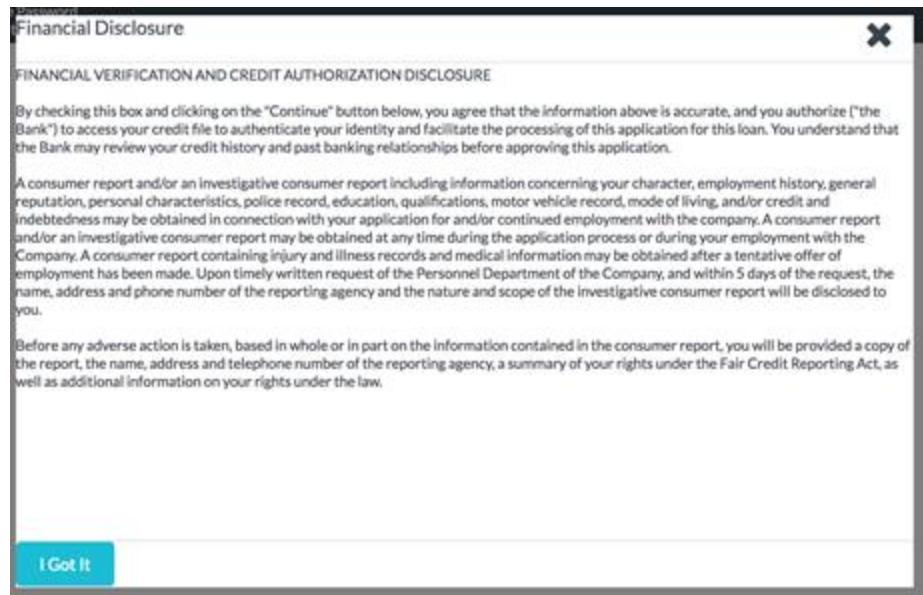
Disclosure Definition Name:

## **Financial Disclosure**

Confirmation message:

I acknowledge and agree to [the Financial Disclosure](#).

```
"disclosures": [{
 "name": "FinancialDisclosure",
 "dialog": {
 "type": "cover",
 "title": "Financial Disclosure",
 "showCloseButton": true,
 "closeText": "I Got It"
 }
}]
```



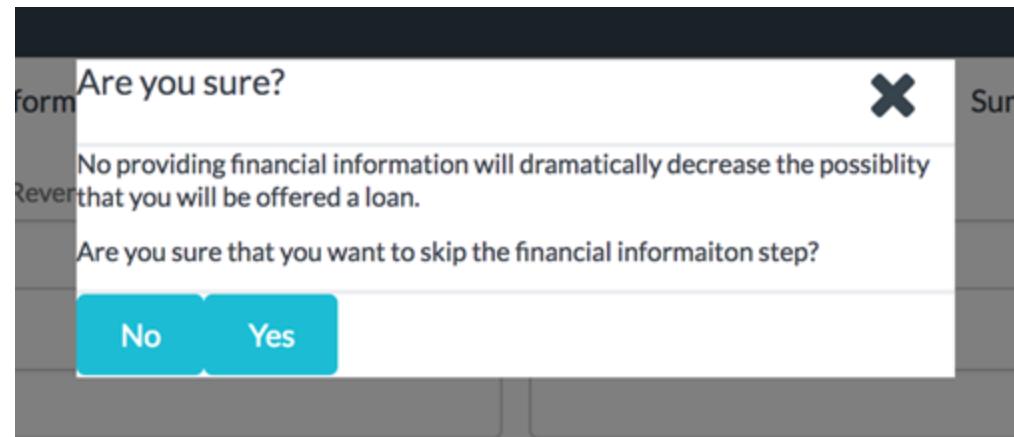
# Confirmation dialog

Certain actions need to be confirmed first, such as deletion, using a modal dialog.

- When users click the confirmation button, the modal dialog is closed and the action is triggered (handler).
- When users click the cancel button, the modal dialog is closed and nothing happens.

I acknowledge and agree to the [Financial Disclosure](#).

[Skip](#) [Save](#)



```
"componentActions": [{
 "name": "cancel",
 "actionType": "normal",
 "label": "Skip",
 "confirmation": {
 "type": "prompt",
 "title": "Are you sure?",
 "content": "<p>No providing financial information will dramatically decrease the possibility that you will be offered a loan.</p><p>Are you sure that you want to skip the financial informaiton step?</p>",
 "showConfirmButtons": true,
 "confirmText": "Yes",
 "cancelText": "No"
 },
 "handler": {
 "assignments": [{
 "field": "$nav(ApplicationSteps).appearance(current);",
 "value": 2
 }]
 }
},
```

## Advanced feature - component with multiple data sources

A component can display data aggregation from multiple SFDC database tables.

- The aggregated data should be treated as normal component data that supports all the portal features, e.g., sorting, grouping, conditional rendering, etc.

How to configure?

1. Each SFDC database table (i.e., each data source) form a data-model component.
2. The aggregated component forms its own data from all the data-model components (as `dataSources`)

```
{
 "name": "SimpleLayout1_PrimaryContentZone",
 "children": [
 {"name": "loanAccountDataModel"},
 {"name": "leaseAccountDataModel"},
 {"name": "collectAccountDataModel"},
 {"name": "accountListView"}
]
}

"loanAccountDataModel": {
 "category": "component",
 "type "property": {"structureName": "loan_account_data_model_v2"}
},
"leaseAccountDataModel": {
 "category": "component",
 "type "property": {"structureName": "lease_account_data_model_v2"}
},
"collectAccountDataModel": {
 "category": "component",
 "type "property": {"structureName": "collect_account_data_model_v2"}
},
```

```
{
 "tableName": "collect_Loan_Account_c",
 "structureName": "collect_account_data_model_v2",
 "sections": [{
 "fields": [
 {"fieldName": "Name", },
 {"fieldName": "collect_Days_Past_Due_c", },
 {"fieldName": "collect_Delinquent_Amount_c", }
]
 }]
}
{
 "tableName": "loan_Loan_Account_c",
 "structureName": "loan_account_data_model_v2",
 "sections": [{
 "fields": [
 {"fieldName": "Name", },
 {"fieldName": "loan_Number_of_Days_Overdue_c", },
 {"fieldName": "collect_Delinquent_Amount_c", }
]
 }]
}
{
 "tableName": "cllease_Lease_Account_c",
 "structureName": "lease_account_data_model_v2",
 "sections": [{
 "fields": [
 {"fieldName": "Name", },
 {"fieldName": "cllease_Days_Past_Due_c", },
 {"fieldName": "cllease_Delinquent_Amount_c", }
]
 }]
}
```

```
{
 "structureName": "account_list_view_v2",
 "sections": [
 "fields": [
 {
 "id": "73e4",
 "fieldName": "Name",
 "fieldUi": {
 "fieldType": "string",
 "label": "Account Name"
 }
 }, {
 "id": "dfaf",
 "fieldName": "PastDue",
 "fieldUi": {
 "fieldType": "string",
 "label": "Past Due"
 }
 }, {
 "id": "ea9c",
 "fieldName": "DelinquentAmount",
 "fieldUi": {
 "fieldType": "currency",
 "label": "Delinquent Amount"
 }
 }]
]
}
```

```
"accountListView": {
 "category": "component",
 "type": "list-view",
 "property": {
 "structureName": "account_list_view_v2",
 "dataSources": [
 "$state(loanAccountDataModel);",
 "$state(leaseAccountDataModel);",
 "$state(collectAccountDataModel);"
],
 "fieldMappings": {
 "$state(loanAccountDataModel)": {
 "Name": "Name",
 "loan_Number_of_Days_Overdue_c": "PastDue",
 "loan_Delinquent_Amount_c": "DelinquentAmount"
 },
 "$state(leaseAccountDataModel)": {
 "Name": "Name",
 "cllease_Days_Past_Due_c": "PastDue",
 "cllease_Delinquent_Amount_c": "DelinquentAmount"
 },
 "$state(collectAccountDataModel)": {
 "Name": "Name",
 "collect_Days_Past_Due_c": "PastDue",
 "collect_Delinquent_Amount_c": "DelinquentAmount"
 }
 }
 },
 "appearance": {
 "columns": 3,
 "sortableFields": [
 "Name",
 "PastDue",
 "DelinquentAmount"
],
 "sortedField": {
 "name": "PastDue",
 "ascending": false
 }
 }
}
```

<sup>①</sup> Account Name	<sup>②</sup> Past Due	<sup>③</sup> Delinquent Amount
PAF CON3	100	
MOR_CON3	100	\$10,000.00
MOR_CON2	50	\$5,000.00
PAF_CON2	50	\$5,000.00
LAI-00000000	50	\$0.00
LAI-00000000	50	\$0.00
MOR_CON1	10	\$1,000.00
PAF_CON1	10	\$1,000.00
LAI-00000001	10	\$0.00
LAI-00000001	10	\$0.00
LES-0000000009	2	\$1,000.00
LES-0000000009	2	\$1,000.00
LES-0000000008	1	\$1,000.00
LES-0000000005	1	\$1,000.00
LES-0000000008	1	\$1,000.00
Loan-1		\$23.00
LAI-00000005		\$0.00
LAI-00000004		\$0.00
LAI-00000003		\$0.00

# Dynamic Actions

In the data aggregation component, records are from multiple data tables. The behavior on each record can be different.

- For example, a list view with accounts from `loan__Loan_Account__c`, `cllease__Lease_Account__c`, and `collect__Loan_Account__c`
- When clicking a record, the corresponding detail view should be open based on the record's source.

Let's form three component JSONs for three detail views:

`basic_contracts_v2.json`, `loan_basic_contracts_v2.json`, and  
`lease_basic_contracts_v2.json`

```
{
 "structureName": "basic_contracts_v2",
 "tableName": "collect_Loan_Account_c",
 "whereClause": {
 "freeForm": " id = ':id'"
 },
}

{
 "structureName": "loan_basic_contracts_v2",
 "tableName": "loan_Loan_Account_c",
 "whereClause": {
 "freeForm": " id = ':id'"
 },
}

{
 "structureName": "lease_basic_contracts_v2",
 "tableName": "cllease_Lease_Account_c",
 "whereClause": {
 "freeForm": " id = ':id'"
 },
}
```

# Dynamic Actions

Then, use `addOnFields` key to add action-specific data to each data model.

```
"loanAccountDataModel": {
 "category": "component",
 "type": "data-model",
 "property": {"structureName": "loan_account_data_model_v2",
 "addOnFields": {"detail_view_structure_name": "loan_basic_contracts_v2"}}}
,
"leaseAccountDataModel": {
 "category": "component",
 "type": "data-model",
 "property": {"structureName": "lease_account_data_model_v2",
 "addOnFields": {"detail_view_structure_name": "lease_basic_contracts_v2"}}}
,
"collectAccountDataModel": {
 "category": "component",
 "type": "data-model",
 "property": {"structureName": "collect_account_data_model_v2",
 "addOnFields": {"detail_view_structure_name": "basic_contracts_v2"}}}
,
```

# Dynamic Actions

`clickRecord` handler can use the `addOnFields` value to dynamically set the `structureName` of the detail view.

```
"generalHandlers": {
 "clickRecord": {
 "assignments": [
 {
 "field": "$nav(accountDetailView).property(structureName);",
 "value": "$args[0].attr(DetailViewStructure);"
 },
 {
 "field": "$nav(accountDetailView).property(whereClause.id);",
 "value": "$args[0].attr(id);"
 }
],
 "next": "DetailPage"
 }
}
```

# Multiple Selection Handlers

```
"generalHandlers": {
 "selectRecord": { handler },
 "deselectRecord": { handler },
 "selectAllRecords": { handler },
 "deselectAllRecords": { handler },
 "selectMultipleRecords": { handler }
}
```

# Popup

When an action's `actionType` is '`context`', it can have the `popup` attribute.

When the action is clicked, a non-blocking popup shows up. The popup can include content and actions defined in the same actor.

Usage: menu button with dropdown options



# Popup Specification

```
"componentActions": [{
 "name": "foo",
 "actionType": "context",
 "popup": {
 "direction": "bottom",
 "anchor": "cursor",
 "width": 200,
 "height": 100,
 "margin": 50,
 "borderColor": "#48ce94",
 "backgroundColor": "#ffffff",

 "content": "Lorem ipsum dolor sit amet consectetur."
 }
}, ...]
```

# Actions in Popup

```
"componentActions": [{
 "name": "foo",
 "actionType": "context",
 "popup": { ... }
}, ... {
 "name": "baz",
 "actionType": "normal",
 "context": "foo"
}]
```

# Pagination

Pagination can be added to list view, queue view, and card view. By default, pagination displays at the top of a component.

Configuration:

- Record line template: \$start; - \$end; of \$total;
- Arrow icons: <<, <, >, >>
- Number of records per page
- Number of page links

# Client-side vs. server-side pagination

## Accounts

1 - 10 of 102 accounts

1 2 3 4 5 6 7 8 9 10 > >>

Account Name	Account Number	Industry	Annual Revenue
--------------	----------------	----------	----------------

```
"appearance": {
 "columns": 4,
 "title": "Accounts",
 "expansionTop": "normal",
 "sortableFields": ["Name", "Industry", "AnnualRevenue"],
 "sortedField": {
 "name": "AnnualRevenue",
 "ascending": false
 },
 "pagination": {
 "recordLineTemplate": "$start; - $end; of $total; accounts",
 "numberOfRecordsPerPage": 10,
 "numberOfPageLinks": 10
 }
}
```

# Sort by

When pagination, when users change the sort-by field, either by clicking a column header in a list view or changing selected values in the sort-by drop-down list in a card view or queue view,

**The data needs to be re-queried from the server.** Therefore, the first page should be highlighted after sorting, no matter which page the user was on.

# V3 New Appearance

```
columns: 3 // three cards in a card view
```

```
fieldColumns: 2 // two-column fields in each card
```

# V3 New Appearance II

Sticky tabs/steps

```
"appearance": {"sticky": true}
```

# V3 New Appearance III

fieldChange handler

```
"generalHandlers": {
 "fieldChange": {
 "field_name "assignments": [...],
 "Updates": [...]
 }
 }
}
```

# V3 New Appearance IV (not in appearance builder yet)

Life-cycle update exceptions - the listed interaction statuses won't be reset.

```
"lifeCycle": {
 "loadData": ...

 "unmount": {
 "Interaction": ["groupCollapseStatus"],

 "dynamicPropertyAndAppearance": true
 }
}
```

# V3 New Appearance V (not in appearance builder yet)

**Static content** can have images from static resource.

```
{
 "backgroundImages": [{
 "name": "csblogo_png",
 "altText": "Company Logo",
 "positionType": "topLeft",
 "position": { "top": 0, "left": 0 },
 "size": { "width": 155, "height": 40 }
 }, {
 "name": "logo2_png",
 "altText": "Company Logo 2",
 "positionType": "topRight",
 "position": { "top": 0, "right": 0 },
 "size": { "width": 128, "height": 128 }
 }]
}
```

- **name** is the static resource **name**.
- **altText** is for accessibility.

# positionType and position

All images are `position: absolute`.

`positionType` has 4 built-in values:

- `"topLeft"`, `"topRight"`, `"bottomLeft"`, `"bottomRight"`

Based on the `positionType` value, `position` has two different attributes to describe the absolute position:

- `"topLeft"` → { `"top": ...`, `"left": ...` }
- `"topRight"` → { `"top": ...`, `"right": ...` }
- `"bottomLeft"` → { `"bottom": ...`, `"left": ...` }
- `"bottomRight"` → { `"bottom": ...`, `"right": ...` }

# V3 New Widget - Slider (not in component builder yet)

```
"fieldUi": {
 "fieldType": "range",
 "rangeOptions": {
 "dataType": "currency",
 "min": 1000,
 "max": 100000,
 "step": 1000
 }
}
```

Loan Amount

\$1,000



\$100,000

# V3 Component Feature (not in component builder yet)

Override global settings

```
"settings": {
 ...
 "locale": "en-us",
 "currency": "USD",
 "minimumFractionDigits},
```

```
 "fieldUi": {
 "fieldType": "range",
 "rangeOptions": {
 "dataType": "currency",
 "min": 1000,
 "max": 100000,
 "step": 1000
 },
 "settingsOptions "minimumFractionDigits }
 }
```

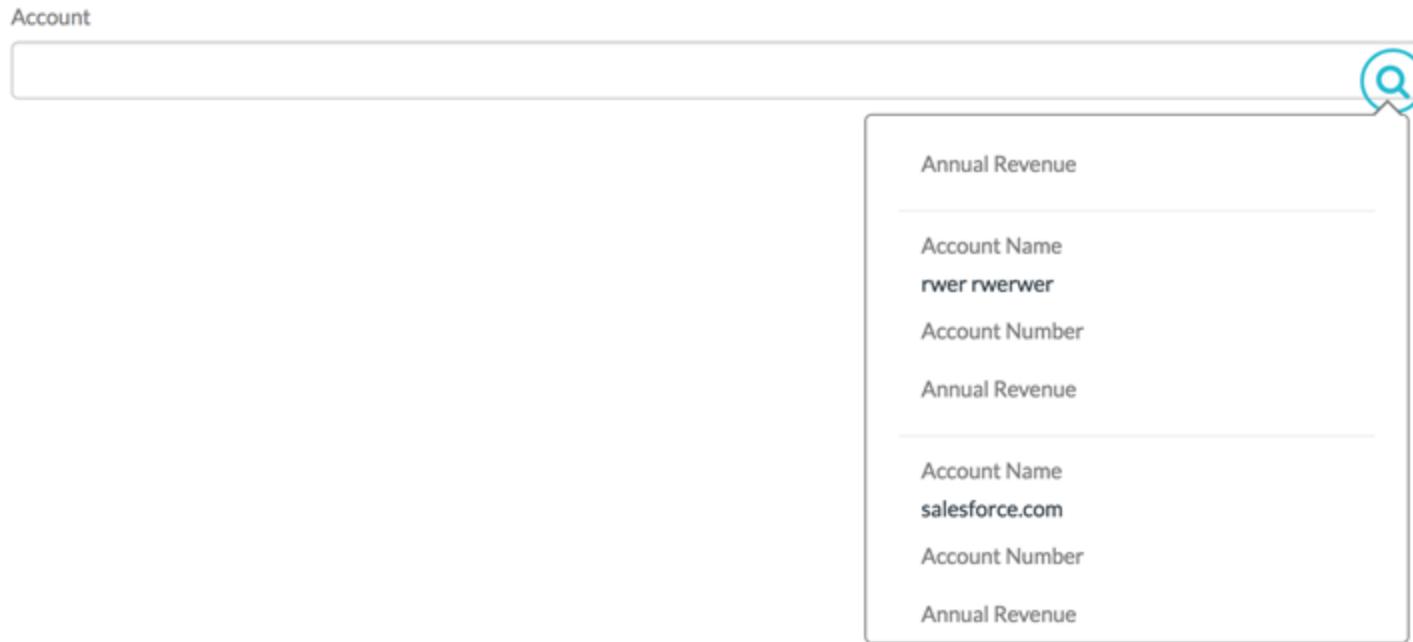
# V3 Component Feature II (not in component builder yet)

For a component field with the `image` `fieldType`, `fieldUi` has an extra attribute of `imageAltText` for accessibility.

```
"fieldUi": {
 "fieldType": "image",
 "linkTemplate":
 "/resource/$global.attr(staticResourceTimestamp);$this.attr(Name);",
 "imageAltText": "$this.attr(Name)"
}
```

# V3 New Feature - Popup (not in portal builder yet)

Any actor can be displayed using popup and can interact with other actors.



# Popup Configuration

```
"AccountPopup": {
 "mode": "popup",
 "category": "component",
 "type": "queue-view",
 ...
 "generalHandlers": {
 "clickRecord": {
 "assignments": [{
 "field": "$state(LoanPanel).attr(account);",
 "value": "$this.attr(Name);"
 }],
 "updates": [{
 "name": "$popup.remove;",
 "value": "AccountPopup"
 }]
 }
 }
}
```

```
 "LoanPanel": {
 ...
 "componentActions": [{
 "name": "searchAccount",
 "actionType": "normal",
 "appearance": {
 "label": "...",
 "style": "..."
 },
 "handler": {
 "updates": [{
 "name": "$popup.add;",
 "value": "AccountPopup"
 }]
 }
 }]
 }
```

ONLY work for detail view or panel right now

# New component field types

- “json”: editable only
- “html”
- “svg”
- “datalist”: read only
- “objectFromJson”

# V3 Navigation Conversion

For queue view and detail view only

columns -> fieldColumns

# V3 Style Conversion - Actor

[name = “foo”] -> [data-style-actor-name = “foo”]

.entry -> [data-style-id = “record”]

.entry .content -> [data-style-id = “record”] . record-content

# V3 Style Conversion

[name = “foo”].section -> [data-style-section-name = “foo”]

# V3 Style Conversion - Field

.field -> [data-style-id = “field”]

[name = “foo”].field -> [data-style-field-name = “foo”]

# V3 Style Conversion - Action

[name = “foo”].action-wrapper -> [data-style-action-name = “foo”]

# V3 Pretty Input

<https://nosir.github.io/cleave.js/>

Formatting features: limited but good enough for now

- Credit card number
- Phone number
- Date
- Numeral
- Custom: blocks, delimiters, prefix

# V3 Pretty Input: Currency Example

```
{
 "fieldName": "Total_Income__c",
 "fieldUi": {
 "fieldType": "currency",
 "label": "Total Income",
 "format": {
 "options": {
 "numeral": true,
 "numeralThousandsGroupStyle": "thousand",
 "numeralDecimalScale": 2
 },
 "prefix": "$"
 }
 }
}
```

# V3 Pretty Input: Percent Example

```
{
 "fieldName": "Net_Worth_of_Business__c",
 "fieldUi": {
 "fieldType": "percent",
 "label": "Net Worth of Business",
 "format": {
 "options": {
 "numericOnly": true,
 "delimiter": ".",
 "blocks": [2, 2]
 },
 "suffix": "%"
 }
 }
}
```

# V3 Pretty Input: SSN Example

```
{
 "fieldName": "SSN_c",
 "fieldUi": {
 "fieldType": "string",
 "label": "Social Security Number",
 "format": {
 "options": {
 "delimiter": "-",
 "numericOnly": true,
 "blocks": [3, 2, 4]
 }
 }
 }
}
```

# Good Practice

Always have the mockup of the complete user interface and the user interaction flow

Detailed annotation (UI specification)

- Layout: division names and actor names
- Fields and sections of each actor
- How to do data query
- Action flow (communication)

# Good Practice II

Use stage prerequisites for

- between-stage communication
- Between-actor communication both actors are not available at the same time

Configure conditional rendering at the end.

- Make everything visible to validate the UI state.

Use save to do incremental configuration.

# Stage Prerequisite

## In Handler

```
"assignments": [{
 "field": "$stage(OfferPage).attr(applicationId);",
 "value": "$result.attr(id);"
},
"next": "OfferPage"
```

## In Stage

```
"property": {
 "structureName": "...",
 "whereClause": {
 "applicationId": "$currentStage.attr(applicationId);"
 }
}
```

# Good Practice III

Use JSON editor as the last resort.

- Action related
- Data structure without backend
- “Special configurations” that are not available through the GUI

## 2.7.1 New Feature I – Stringify objects

Submit action from portal can have data from actors as individual remoteActionArgument, for example

```
"remoteActionArguments": {
 "account": "$this[0];", // object from the first record
 of the current actor
 "accountList": "$state(AccountListView);" // array of
 objects from the AccountListView actor
}
```

\* Data in an actor is always in the array format, even the actor is a detail view, whose data is a single-element array.

## 2.7.1 New Feature II – Dynamic Tabs/Steps

Tabs can be conditionally rendered and enabled; steps can be conditionally rendered. (Not in the actor definition panel yet)

```
{ "current": 1,
 "titles": [
 "Foo",
 {"title": "Bar", "render": "$state(SampleStepControl).attr(hideBar); != true"},

 {"title": "Baz", "render": "$state(SampleStepControl).attr(hideBaz); != true"},

 {"title": "Qux", "render": "$state(SampleStepControl).attr(hideQux); != true"},

 {"title": "Corge", "render": "$state(SampleStepControl).attr(hideCorge); != true"},

 {"title": "Waldo", "render": "$state(SampleStepControl).attr(hideWaldo); != true"}

]
}

{ "titles": ["Wibble", "Wobble",
 {"title": "Wubble",
 "render": "$state(SampleTabControl).attr(hideWubble); != true",
 "enable": "$state(SampleTabControl).attr(disableWubble); != true"},

 {"title": "Flob",
 "render": "$state(SampleTabControl).attr(hideFlob); != true",
 "enable": "$state(SampleTabControl).attr(disableFlob); != true"}

]
}
```

## 2.7.1 New Feature II (cont.)

Two special step values can be assigned to `current`: `next` and `previous`

- `next`: go to the next visible step. If the current step is the last visible step, stay at the current step.
- `previous`: go to the previous visible step. If the current step is the first visible step, stay at the current step.

```
"assignments": [
 {"field": "$nav(SampleSteps).appearance(current);",
 "value": "previous"}
]
```

```
"assignments": [
 {"field": "$nav(SampleSteps).appearance(current);",
 "value": "next"}
]
```

## 2.7.1 New Feature III – Mobile Detecting

Cross-platform in production VF page

```
<apex:page ...>
 ...
 <html>
 <head>
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
 ...
 <script src="https://cdnjs.cloudflare.com/ajax/libs/mobile-
detect/1.4.3/mobile-detect.min.js"></script>
 </head>
 <body>
 ...
 <script type="text/javascript" src="https://.../portal.min.js"></script>
 <link rel="stylesheet" type="text/css" href="https://.../portal.css"/>
 </body>
 </html>
</apex:page>
```

## 2.7.1 New Feature III (cont.) - Automation Initial Data Seed

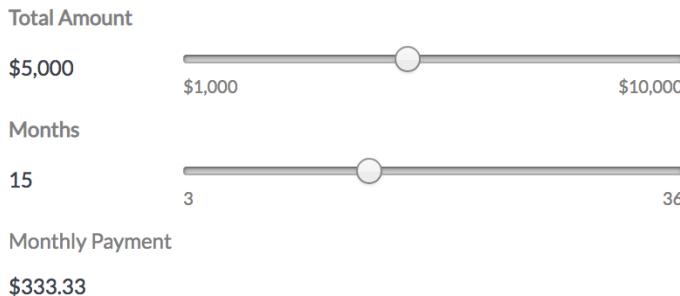
Default navigation JSON and default style JSON are included in builder.js.

Users do not need to seed the `navigationstructure` table and the `css_configuration` table for new applications.

## 2.7.1 New Feature IV – Dynamic Calculation

Value can be dynamically calculated using `fieldChange` handler and `mathjs`.

```
@mathjs{...} ;
```



```
generalHandlers:
{
 "fieldChange": {
 "totalAmount, months": {
 "assignments": [{
 "field": "$this.attr(monthlyAmount);",
 "value":
 "@mathjs{ $this.attr(totalAmount) /
 $this.attr(months) ; } ;"
 }]
 }
 }
}
```

## 2.7.1 Dynamic Calculation (cont.)

Principal	Annual Interest
<input type="text" value="100"/>	<input type="text" value="0.18"/>
Compounded per Year	Years
<input type="text" value="10"/>	<input type="text" value="4"/>
Amount	
\$204.13	

Compound Interest Formula (including Principal)

$$A = P \left(1 + \frac{r}{n}\right)^{nt}$$

Amount  
Interest Rate (decimal)  
Time (years)

Principal  
Number of times interest is compounded per year

thecalculatorsite.com

```
generalHandlers
{
 "fieldChange": {
 "principal, annualInterest,
 compoundTimes, years": {
 "assignments": [
 {
 "field": "$this.attr(amount);",
 "value": "@mathjs{$this.attr(principal); * ((1 +
 $this.attr(annualInterest); /
 $this.attr(compoundTimes);) ^
 ($this.attr(compoundTimes); *
 $this.attr(years));});"
 }
]
 }
 }
}
```

## 2.7.1 New Feature V – Keyword Filter

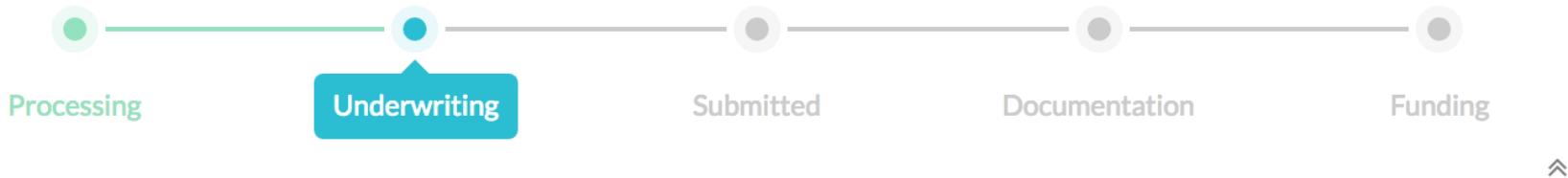
### Keyword filter

```
appearance: {
 "filters": {
 "keyword": {
 "filterLabel": "Filter by Account Name",
 "fieldNames": ["Name"]
 }
 }
}
```

Filter by Account Name

Account Name	Industry	Annual Revenue
GenePoint	Biotechnology	\$30,000,000.00
United Oil & Gas, UK	Energy	
United Oil & Gas, Singapore	Energy	
Edge Communications	Electronics	\$139,000,000.00

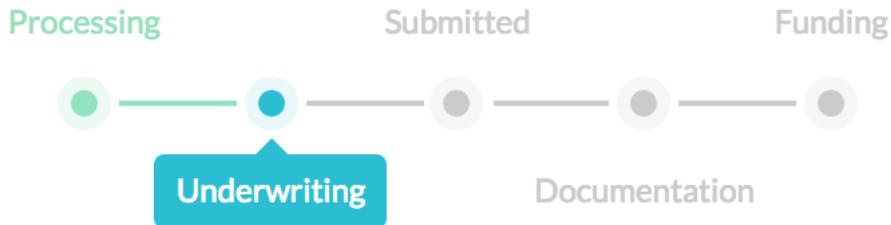
## 2.7.1 New Feature VI – Multi-Value Status Indicator (MVI)



A **read-only** picklist can be rendered as multiValueIndicator

```
"fieldName": "Application_Status",
"fieldUi": {
 "fieldType": "multiValueIndicator",
 "defaultValue": "Underwriting",
 "selectOptions": [
 {
 "optionText": "Processing",
 "optionValue": "Processing",
 }, ...
]
}
```

## 2.7.1 MVSI (cont.)



Compact view for narrow horizontal space  
(e.g., phone)

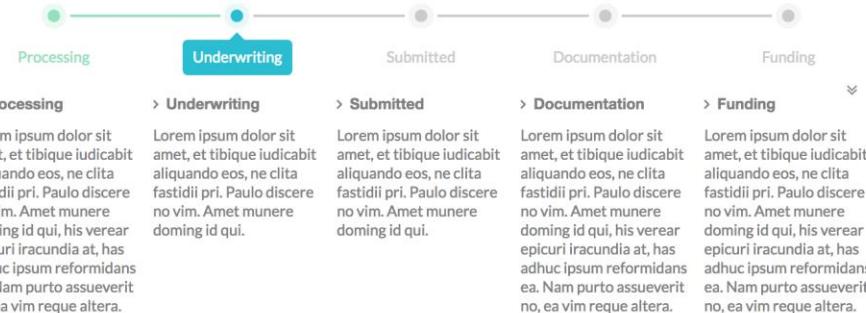
The current highlighted status is always displayed below the dotted line.

```
"fieldName": "Application_Status",
"fieldUi": {
 "fieldType": "multiValueIndicator",
 "defaultValue": "Underwriting",
 "selectOptions": [
 {
 "optionText": "Processing",
 "optionValue": "Processing",
 }, ...
],
 "format": { "compact": true }
}
```

## 2.7.1 MVSI (cont.)

The indicator can be configured to be expandable for extra information of each value.

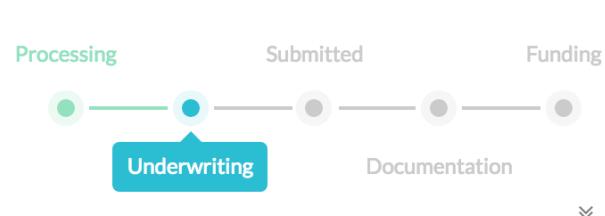
```
"fieldUi": {
 "fieldType": "multiValueIndicator",
 "expansible": true,
 "expansibleIcons": {
 "expanded": "",
 "collapsed": ""
 },
 "selectOptions": [{
 "optionText": "Processing",
 "optionValue": "Processing",
 "optionExpansionContent": "..."
 }, ...],
```



## 2.7.1 MVSI (cont.)

Expanded view in the compact mode

Expanded content is in paragraph. The information about the current highlighted status is indented.



### > Processing

Lorem ipsum dolor sit amet, et tibique iudicabit aliquando eos, ne clita fastidii pri. Paulo discere no vim. Amet munere doming id qui, his verear epicuri iracundia at, has adhuc ipsum reformidans ea. Nam purto assueverit no, ea vim reque altera.

### > Underwriting

Lorem ipsum dolor sit amet, et tibique iudicabit aliquando eos, ne clita fastidii pri. Paulo discere no vim. Amet munere doming id qui.

### > Submitted

Lorem ipsum dolor sit amet, et tibique iudicabit aliquando eos, ne clita fastidii pri. Paulo discere no vim. Amet munere doming id qui.

### > Documentation

Lorem ipsum dolor sit amet, et tibique iudicabit aliquando eos, ne clita fastidii pri. Paulo discere no vim. Amet munere doming id qui, his verear epicuri iracundia at, has adhuc ipsum reformidans ea. Nam purto assueverit no, ea vim reque altera.

### > Funding

Lorem ipsum dolor sit amet, et tibique iudicabit aliquando eos, ne clita fastidii pri. Paulo discere no vim. Amet munere doming id qui, his verear epicuri iracundia at, has adhuc ipsum reformidans ea. Nam purto assueverit no, ea vim reque altera.

## 2.8 New Feature I – Visible Tab/Step Indexes

```
{
 "titles": [
 "Foo",
 { "title": "Bar", "render": "$state(StepControl).attr(hideBar); != true" },
 { "title": "Baz", "render": "$state(StepControl).attr(hideBaz); != true" },
 "Qux"
],
 "visibleIndexes": ["I", "II", "III", "IV"]
}
```

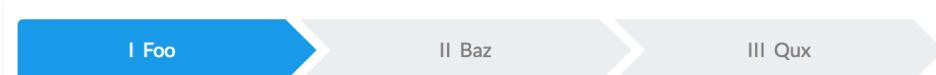


Hide Bar

Hide Bar

Hide Baz

Hide Baz



Hide Bar

Hide Bar

Hide Baz

Hide Baz

## 2.8 New Feature II – Format number field

- The number field is now formatted into locale string.

Account Name	Industry	Annual Revenue	Email	Employees
GenePoint	Biotechnology	\$30,000,000.00		265
United Oil & Gas, UK	Energy			24,000
United Oil & Gas, Singapore	Energy			3,000
Edge Communications	Electronics	\$139,000,000.00		1,000
Burlington Textiles Corp of Ame...	Apparel	\$350,000,000.00		9,000
Pyramid Construction Inc.	Construction	\$950,000,000.00		2,680
Dickenson plc	Consulting	\$50,000,000.00		120

## 2.8 New Feature III – Dynamic current for Tabs/Steps

- `current` can be set from url or internal state

`https://c.na53.visual.force.com/apex/TaurusM2Test?startStep=3`

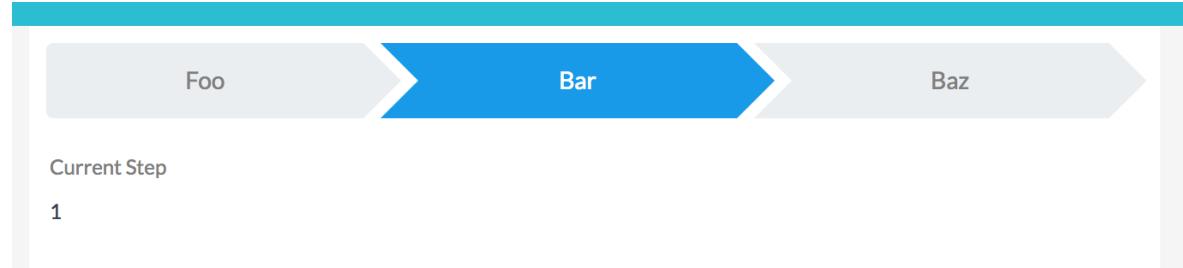
```
"global": {
 "startStep": "$url(startStep);"
}
"actors": {
 "Steps": {
 "category": "container",
 "type": "steps",
 "appearance": {
 "current": "$global.attr(startStep);",
 ...
 }
 }
}
```



## 2.8 Dynamic current (cont.)

```
{
 "current": "$state(Steps123Control).attr(currentStep);",
 "titles": [...]
}

{
 "structureName": "...",
 "sections": [{
 "fields": [{
 "id": "764a",
 "fieldName": "currentStep",
 "fieldUi": {
 "fieldType": "number",
 "label": "Current Step",
 "defaultValue": 1
 }
 }]
 }]
}
```



## 2.8 New Feature IV – Handler Triggering Actions

- Handler from one action can conditionally trigger other actions (in the same or different actors)
  - The triggered action can be rendered false.
  - **But the actor has to be in the current stage.**

```
successHandler: {
 "assignments": [...]
 "updates": [...]
 "cruds": [...]
 "actions": [
 { "condition": "$result...", "actorName": ..., "name": ... }
]
}
```

```
"AccountDetails2": {
 "componentActions": [{
 "name": "save",
 "actionType": "save",
 "successHandler": {
 "assignments": [{
 "field": "$state(AccountDetail2)",
 "value": "$result;"
 }],
 "actions": [{
 "actorName": "ContactList",
 "name": "saveContacts"
 }]
 }
 }],
}

"ContactList": {
 "componentActions": [{
 "name": "saveContacts",
 "actionType": "save",
 "appearance": { "render": false },
 "remoteActionArguments": { "AccountId": "$state(AccountDetail2).attr(id); " },
 }],
}
```

The screenshot shows a user interface for managing account details and contact lists. At the top right is a green button labeled 'Save Account'. Below it is a section titled 'AccountDetail2' containing fields for 'Account Name' (with a required red asterisk), 'Industry', 'Annual Revenue', and 'Employees'. A horizontal line separates this from another section titled 'ContactList' at the top right. This section contains fields for 'First Name' (required), 'Last Name' (required), 'Business Phone', and 'Email'. There is also a green 'Add' button and a small 'x' icon to remove items from the list.

## 2.8 New Feature V – Google Place Autocomplete

- Get Google API key
- Add the URL to the whitelist of Google API key
- Add the following line to the VF page

```
<script
src="https://maps.googleapis.com/maps/api/js?key=your+key&li
braries=places"></script>
```

```
{"fields": [{}
 "id": "31ad", "fieldName": "address",
 "fieldUi": {
 "fieldType": "addressAutocomplete",
 "addressComponentMapping": [{}
 "addressComponent": "street_number", "fieldName": "address"
 }, {}
 "addressComponent": "route", "fieldName": "address"
 }, {}
 "addressComponent": "locality", "fieldName": "city"
 }, {}
 "addressComponent": "administrative_area_level_1", "fieldName": "state"
 }, {}
 "addressComponent": "country", "fieldName": "country"
],
 "addressComponentRestrictions": {"country": ["in"]}
 }
,
 {"id": "2bd5", "fieldName": "city", "fieldUi": { "fieldType": "string", "label": "City" }},
 {"id": "daeb", "fieldName": "state", "fieldUi": { "fieldType": "string", "label": "State" }},
 {"id": "27a0", "fieldName": "country", "fieldUi": { "fieldType": "string", "label": "Country" }}
]...
• addressComponentRestrictions is optional.
• addressComponent types can be found at
https://developers.google.com/maps/documentation/geocoding/intro#Types
```

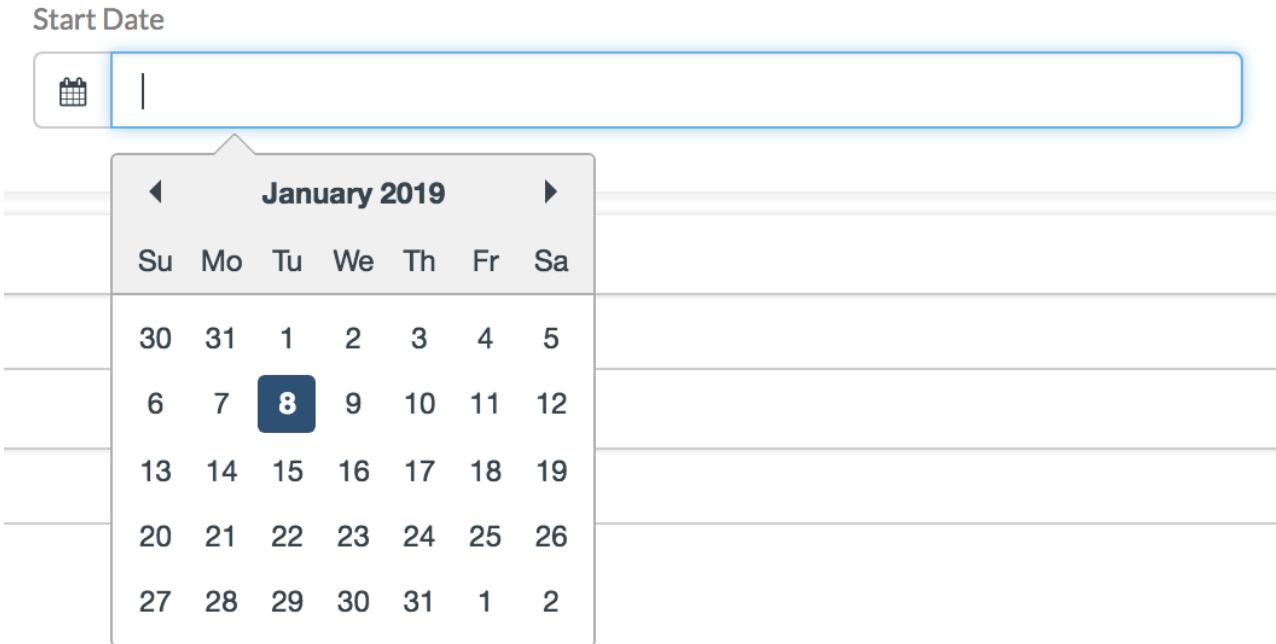
## 2.8 Google Place Autocomplete

- "formatted\_address" is a special address component, including the complete suggested address

```
"fieldUi": {
 "fieldType": "addressAutocomplete",
 "addressComponentMapping": [
 { "addressComponent": "formatted_address", "fieldName": "Home_Address" },
 { "addressComponent": "street_number", "fieldName": "Street_Number" },
 { "addressComponent": "route", "fieldName": "Street_Name" },
 { "addressComponent": "locality", "fieldName": "City" },
 { "addressComponent": "administrative_area_level_1", "fieldName":
 "State_Address" },
 { "addressComponent": "postal_code", "fieldName": "Postal_Code" }
 "addressComponentRestrictions": { "country": ["aus"] }
}
```

## 2.8 New Feature VI – Date Picker (Deprecated)

```
"fieldUi": { "fieldType": "datepicker", "label": ... }
```



## 2.8 Date Picker (cont.) (Deprecated)

```
"fieldUi": {
 "fieldType": "datepicker",
 "placeholderText": "Click to select a date",
 "format": {
 "options": { "datePattern": "dd/MM/yyyy" } ...
 }
}
```

Default datePattern: MM/dd/yyyy

Month: M (e.g., 1), MM (01), MMM (Jan), MMMM (January)

Day: d (e.g., 1), dd (01)

Year: yy (e.g., 19), yyyy (2019)

## 2.8 Date Picker (cont.) (Deprecated)

- Tested date format
  - yyyy-MM-dd
  - MM/dd/yyyy
  - dd/MM/yyyy

## 2.8 Date Picker (cont.) (Deprecated)

- Date picker by default has a wide range. For example, you can add a date in year 5555.
- You can specify `startYear` and `endYear` to configure the year range.

```
"options": { "startYear": 1969, "endYear": 2050 },
```

## 2.8 Date Picker (cont.) (Deprecated)

You can set `dateLocale` in global settings "dateLocale": "en-US"

Available `dateLocales`:

zh-CN, Chinese Simplified

nl, Dutch (Netherlands)

en-CA,

en-US (default),

fi, Finnish

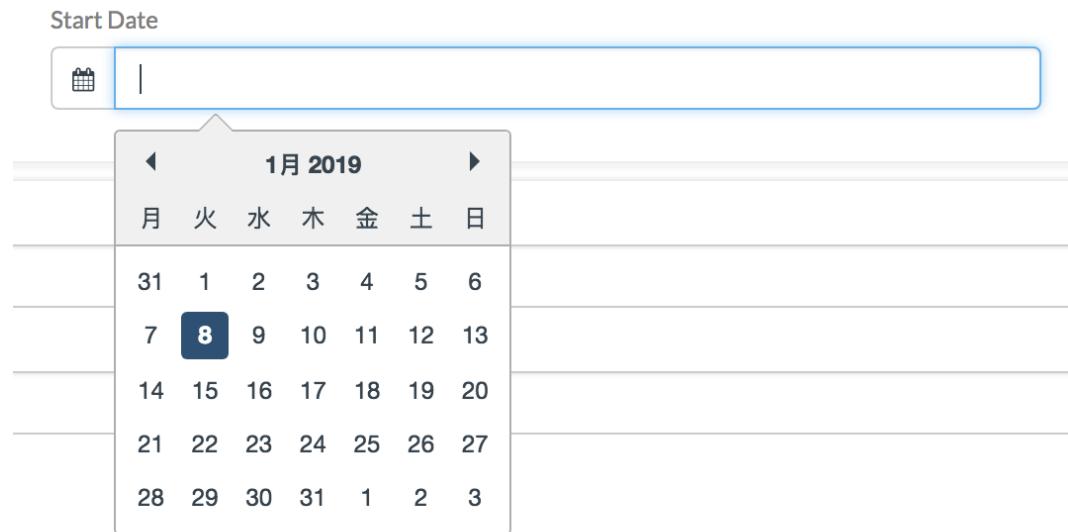
fr, French

de, German

it, Italian

ja, Japanese

ru, Russian



## 2.8 New Feature VII – Multi-value Picklist

```
"fieldUi": {
 "fieldType": "multipicklist",
 "selectOptions": [...]
}
```

Same selectOptions as picklist

Supported in the date query dialog

- Foo Bar \*
- Apple
  - Orange
  - Grape
  - Strawberry
  - Banana
  - Pear
  - Watermelon
  - Pineapple

Display Type

Data Type

MULTIPICKLIST

Display Type \*

Multiple-Selected Picklist

Text

Label

Placeholder Text

Foo Bar

## 2.8 No more communityPrefix in NavigationJSON

- CommunityPrefix is read from the backend and stored in userInfo.

2.8 More third-party JS libraries are removed from the bundle `portal.min.js` and `builder.min.js`

```
<head>
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
 <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Lato&subset=latin,latin-ext"/>
 <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro"/>
 <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"/>
 <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"/>
</head>
<body>
 <script src="../../soap/ajax/29.0/connection.js" type="text/javascript"></script>
 <script src="../../soap/ajax/29.0/apex.js" type="text/javascript"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/mathjs/3.13.3/math.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/ace/1.2.8/ace.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/dompurify/1.0.4/purify.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/mobile-detect/1.4.3/mobile-detect.min.js"></script>
 <script crossorigin="anonymous" src="https://unpkg.com/react@16/umd/react.production.min.js"></script>
 <script crossorigin="anonymous" src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/immutable/3.8.2/immutable.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/redux/4.0.0/redux.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/react-redux/5.0.7/react-redux.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.22.1/moment.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/axios/0.18.0/axios.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/node-uuid/1.4.8/uuid.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/redux-thunk/2.2.0/redux-thunk.min.js"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/react-transition-group/2.2.1/react-transition-group.min.js"></script>
</body>
```

## 2.7.1 to 2.8 DOM Changes

- <div class="control"><div class="input-main">...
- <div class="entries"><div class="record-container"><div data-style-id="record">...

# Portal Utility

- cd to src/utility
- node builder-loader.js <username> <password> <securityToken>
  - To load the navigation JSONs, component JSONs, and default layouts under src/applications/editor to a specific SFDC org.
  - Security token is optional.
  - For example
    - node builder-loader.js min@portaltraining.com merc1235
    - XF7Rx0NGnnRWuJ3izSZhG5S1

# Portal Utility (cont.)

- node builder-loader.js <username> <password>  
<securityToken> **true**
  - For sandbox
- node builder-cleaner.js
  - To check unused builder-related component JSONs that are not referred by any builder-related navigation JSON.

# Portal Utility (cont.)

- The following 2 converters convert the branding configuration of existing portal applications from 2.7 to 2.8, which will be moved to VF page eventually.
- `cd to src/utility`
- `node builder-converter-27-to-271.js <username> <password> <securityToken>`
- `node builder-converter-271-to-28.js <username> <password> <securityToken>`

## 2.8.1 New Feature I – Range Enhancement

- 1. Range value is editable and supports format, prefix, and suffix

```
{ "fieldName": "loanAmount",
 "fieldUi": {
 "fieldType": "range",
 "label": "Loan Amount",
 "rangeOptions": {
 "dataType": "currency",
 "min": 1000, "max": 10000, "step": 10,
 "editableRangeValue": true
 },
 "settingsOptions": { "minimumFractionDigits": 0 },
 "format": {
 "options": { "numeral": true, "numeralThousandsGroupStyle": "thousand",
 "numeralDecimalScale": 0 },
 "prefix": "$"
 } } }
```

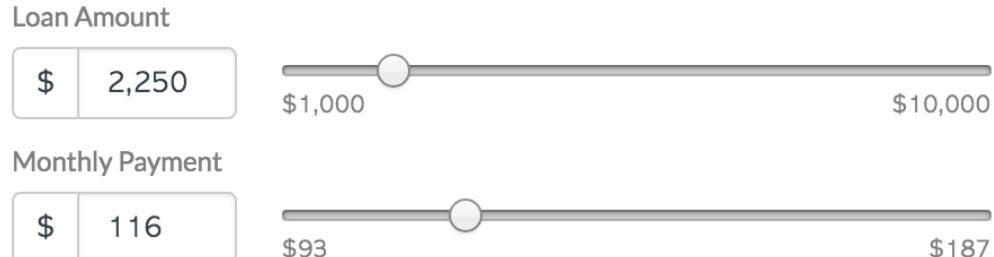
Loan Amount



## 2.8.1 Range Enhancement (cont.)

- Range min, max, step can be dynamically set by mathjs and variables.

```
{ "fieldName": "loanAmount", "fieldUi": { ... } },
{
 "fieldName": "payment",
 "fieldUi": {
 "fieldType": "range",
 "label": "Monthly Payment",
 "rangeOptions": {
 "dataType": "currency",
 "min": "@mathjs{$this.attr(loanAmount) / 24};",
 "max": "@mathjs{$this.attr(loanAmount) / 12};",
 "editableRangeValue": true
 },
 "settingsOptions": { "minimumFractionDigits": 0 },
 "format": { ... }
 }
}
```



## 2.8.1 Range Enhancement (cont.)

- tickMarks

Market Value

\$ 800,000

Loan Amount

\$ 108,000



```
"fieldUi": {
 "fieldType": "range", "label": "Loan Amount",
 "rangeOptions": {
 "dataType": "currency",
 "min": 75000, "max": "@mathjs{$this.attr('marketValue') * 0.7};",
 "step": 1000, "editableRangeValue": true
 },
 "tickMarks": [
 {"text": "Market Value", "position": 0},
 {"text": "50%", "position": "@mathjs{($this.attr('marketValue') * 0.5 - 75000) / ($this.attr('marketValue') * 0.7 - 75000)}"},
 {"text": "60%", "position": "@mathjs{($this.attr('marketValue') * 0.6 - 75000) / ($this.attr('marketValue') * 0.7 - 75000)}"},
 {"text": "70%", "position": 1}
],
 "settingsOptions": { "minimumFractionDigits": 0 },
 "format": {
 "options": { "numeral": true, "numeralThousandsGroupStyle": "thousand",
 "numeralDecimalScale": 0 },
 "prefix": "$"
 }
}
```

Tickmark's position is a floating-point number between 0 and 1, inclusively.  
Tickmark with position < 0 or > 1 won't get rendered.

## 2.8.1. Caused Style Issue

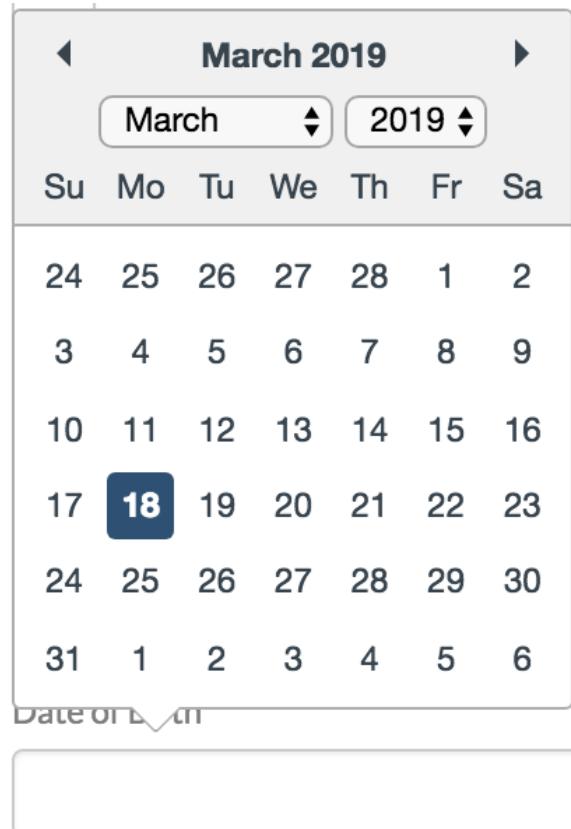
- .input-name now wraps .input-prefix and .input-suffix

```
<div class="input-main with-suffix">
 <input id="68kx4rfig9h" class="form-control" type="number" value="1">
 px
</div>
```

## 2.8.1 New Feature II – Datepicker Enhancement

- Datepicker field now can accept all the options supported by [react-datepicker](#).

```
"fieldUi": {
 "fieldType": "datepicker",
 "label": "Date of Birth",
 "format": {
 "options": {
 "datePattern": "yyyy-MM-dd",
 "minYear": 1970,
 "showMonthDropdown": true,
 "showYearDropdown": true,
 "dropdownMode": "select"
 }
 }
}
```



## 2.8.1 New Feature III – Multiple-File Upload

- Upload action can be configured so that when users click a Upload button to open the file dialog, they can select multiple files to upload them at the same time.

```
{
 "name": "upload",
 "actionType": "upload",
 "property": {
 "multiple": true
 },
 "appearance": {
 "label": "Upload",
 "style": "primary"
 },
}
```

## 2.8.1 Multiple-File Upload With Drag and Drop

- Users can also drag files from desktop and drop them to a "droppable" component or entry to upload. (Of course, there must be an Upload button associated with the component or entry.)

```
"appearance": {
 "droppableComponent": {
 "uploadActionName": "upload",
 "type": "FILE"
 }
}
```

```
"componentActions": [{
 "name": "upload",
 "actionType": "upload",
 ...}
]
```

```
"appearance": {
 "droppableRecord": {
 "uploadActionName": "upload",
 "type": "FILE"
 }
}
```

```
"recordActions": [{
 "name": "upload",
 "actionType": "upload",
 ...}
]
```

## 2.8.1 New Feature IV – Multi-Step Calculation

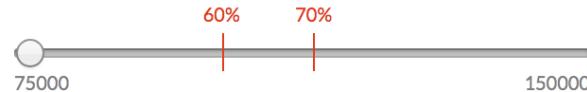
- Multi-step calculation can be used for complicated loan calculations (in assignments ONLY).
- Each step is a calculation assigned to an intermediate variable, which can be used by the following steps.
  - To avoid name conflict, the intermediate variable's name should NOT **start with var**, which is internally used by portal framework.
  - The variable name of the last step must be **final**, which holds the final calculated value.

```
"assignments": [{
 "field": "$this.attr(totalAmount);",
 "mathValue": [
 "x = ...",
 "y = ...",
 "final = ..."
] }]
```

## 2.8.1 Multi-Step Calculation Example

Loan Amount

75000



Interest (50%)

4.2% (Variable)

Total Amount

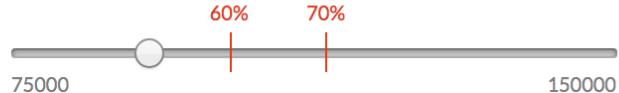
\$78,150.00

- Users can specify a loan amount between 75000 and 150000.
- Interest changes with different loan amount.
- Total amount = Loan Amount \* (1 + selected interest rate)

	variable	1 year	3	5	8	10
Loan amount < 100000 (50%)	4.2	4.4	4.4	4.5	4.8	5
Loan amount < 120000 (60%)	4.4	4.6	4.6	4.7	5	5.2
Loan amount < 140000 (70%)	4.6	4.8	4.8	4.9	5.2	5.4

Loan Amount

91000



Total Amount

\$94,822.00

Interest (60%)

✓ 4.2% (Variable)

4.4% (1 Year)

**4.4% (3 Years)**

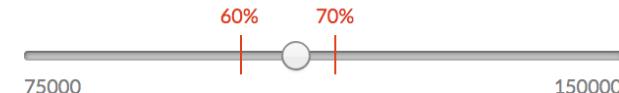
4.5% (5 Years)

4.8% (8 Years)

5.0% (10 Years)

Loan Amount

109000



Total Amount

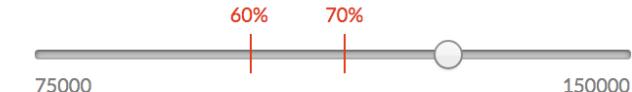
\$114,014.00

Interest (60%)

4.6% (3 Years)

Loan Amount

128000



Total Amount

\$134,272.00

Interest (70%)

4.9% (5 Years)

```
"fields": [{}
 "fieldName": "loanAmount",
 "fieldUi": {
 "fieldType": "range", "label": "Loan Amount",
 "rangeOptions": { "min": 75000, "max": 150000, "step": 1000 },
 "tickMarks": [{
 "text": "60%",
 "position": "@mathjs{ (120000 - 75000) / (200000 - 75000) };"
 }, {
 "text": "70%",
 "position": "@mathjs{ (140000 - 75000) / (200000 - 75000) };"
 }]
 },
 {}},

{ "fieldName": "interest50",
 "fieldUi": {
 "fieldType": "picklist", "label": "Interest (50%)", "defaultValue": "variable",
 "selectOptions": [
 { "optionText": "4.2% (Variable)", "optionValue": "variable" },
 { "optionText": "4.4% (1 Year)", "optionValue": "1y" },
 { "optionText": "4.4% (3 Years)", "optionValue": "3y" },
 { "optionText": "4.5% (5 Years)", "optionValue": "5y" },
 { "optionText": "4.8% (8 Years)", "optionValue": "8y" },
 { "optionText": "5.0% (10 Years)", "optionValue": "10y" }
] }},
{ "fieldName": "interest60", ... },
{ "fieldName": "interest70", ... }
```

```
"appearance": {
 "renderFields": {
 "interest50": "($this.attr(loanAmount) / 200000) < 0.5",
 "interest60": "($this.attr(loanAmount) / 200000) >= 0.5 and ($this.attr(loanAmount) / 200000) < 0.6",
 "interest70": "($this.attr(loanAmount) / 200000) >= 0.6"
 }
}

"generalHandlers": { "fieldChange": {
 "loanAmount, interest50, interest60, interest70": {
 "assignments": [{}],
 "field": "$this.attr(totalAmount);",
 "mathValue": [
 "i50 = $this.attr(interest50);",
 "i60 = $this.attr(interest60);",
 "i70 = $this.attr(interest70);",
 "r50 = i50 == \"variable\" ? 4.2 : (i50 == \"1y\" ? 4.4 : (i50 == \"3y\" ? 4.4 : (i50 == \"5y\" ? 4.5 : (i50 == \"8y\" ? 4.8 : 5.0))))",
 "r60 = i60 == \"variable\" ? 4.4 : (i60 == \"1y\" ? 4.6 : (i60 == \"3y\" ? 4.6 : (i60 == \"5y\" ? 4.7 : (i60 == \"8y\" ? 5.0 : 5.2))))",
 "r70 = i70 == \"variable\" ? 4.6 : (i70 == \"1y\" ? 4.8 : (i70 == \"3y\" ? 4.8 : (i70 == \"5y\" ? 4.9 : (i70 == \"8y\" ? 5.2 : 5.4))))",
 "perct = $this.attr(loanAmount) / 200000",
 "rate = perct < 0.5 ? r50 : (perct < 0.6 ? r60 : r70)",
 "final = $this.attr(loanAmount) * (1 + rate/100)"
]
 }
}] }
```

## 2.9 IncludePortal2 component configuration

```
<apex:page showHeader="false" sidebar="false" standardStylesheets="false" applyHtmlTag="false" applyBodyTag="false"
docType="html-5.0" controller="clcommon.PortalActions">
 <apex:include pageName="clcommon__ResolveNamespace"/>

 <clcommon:IncludePortal2 navName="..." theme="..." showDefaultLoadingPage="...">
 </clcommon:IncludePortal2 >

</apex:page>
```

**navName (required):** NavigationStructure Name

**theme (optional, default='mint', selection:["mint", "mui"]):** CSS themes

**showDefaultLoadingPage (optional, default="true", selection:["true", "false"]):** Boolean to  
show/hide default loading page

## 2.9 Portal Enhancements

1. Currency symbol
2. Negative boolean

## 2.9 Portal Enhancement I – Currency Symbol

- For the currency field, Portal will use `currencySymbol` in the global settings and display the symbol as the field prefix.
- `currency` in the global settings becomes deprecated.

The screenshot shows a portal page with three fields:

- Annual Revenue:** A dropdown menu with options: £ 5,470,000,000.00 (selected), £ 500,000,000.00, and £ 500,000,000.00.
- Account Phone:** An input field containing (212) 842-5500.
- Industry:** A dropdown menu with option: Energy.

```
"settings": {
 "currencySymbol": "£"
}
```

## 2.9 Currency Symbol (cont.)

- The global settings can indicate that the currency symbol is a suffix.

```
"settings": {
 "currencySymbol": "€",
 "currencySymbolAsSuffix": true
}
```

The screenshot shows a user interface with two main sections. On the left, there is a sidebar with a dark blue header containing the text 'Annual Revenue' with a downward arrow icon. Below this are three items: '5,470,000,000.00 €', '500,000,000.00 €', and '500,000,000.00 €'. On the right, there is a main content area with a light gray background. It contains a form with several fields:

- A 'Phone' field labeled 'Account Phone' containing '(212) 842-5500'.
- An 'Industry' field containing 'Energy' with a dropdown arrow.
- An 'Annual Revenue' field containing '5,470,000,000' followed by a right-aligned '€' symbol.

The 'Annual Revenue' field has a red border, indicating it is the active or selected field. The entire interface has a clean, modern design with a white background and light gray borders for the sections.

## 2.9 Currency Symbol (cont.)

- At the field level (as currency), users can override the global settings.

```
// prefix does not display
"fieldUi": {
 "fieldType": "currency",
 "format": { "options": {...}, "prefix": false }
}
```

```
// custom prefix string
"fieldUi": {
 "fieldType": "currency",
 "format": { "options": {...}, "prefix": "..."}
}
```

```
// same as suffix
```

## 2.9 Portal Enhancement II – Negative Boolean

- If a field type is `-boolean`, it is rendered as a checkbox, which is checked if its value is `false` and unchecked if its value is `true`.

## 2.9 Portal Modularization

1. Remove cross-actor dependency
2. Step enhancements
3. Cross-actor validation
4. Query with default new

## 2.9 Portal Modularization I - \$currentStage

- Actors can get value from the current stage prerequisites.
- Actors can set value to the current stage prerequisites through assignments

```
"property": {
 "structureName": "account_detail_view",
 "whereClause": { "accountId": "$currentStage.attr(highlightedAccountId); " }
}
"generalHandlers": {
 "clickRecord": {
 "assignments": [{
 "field": "$currentStage.attr(highlightedAccountId); ",
 "value": "$args[0].attr(id); "
 }]
 } }
"successHandler": {
 "assignments": [{
 "field": "$currentStage.attr(highlightedAccountId); ",
 "value": "$result.attr(id); "
 }],
```

## 2.9 Portal Modularization I - autoHighlightedId

- List / queue / card views may have `autoHighlightedId`, which supports variables.

```
"appearance": {
 ...
 "autoHighlightedId": "$currentStage.attr(highlightedAccountId);",
}
```

## 2.9 Portal Modularization II – Steps / Tabs Enhancements

- Steps / Tabs may have unique names.
- Steps / Tabs may have componentActions, which may belong to specific steps/tabs.

The screenshot shows a modular form with three steps. Step 1: Business Information contains fields for Business Name (Foo) and Annual Revenue (\$). Step 2: Business Address contains an Account Number field. Step 3: Summary contains a My New PickList field. A red arrow points from the bottom 'NEXT' button to the 'NEXT' button on the right side of the third step.

```
"appearance": {
 "titles": [{
 "title": "...",
 "name": "info"
 }, {
 "title": "...",
 "name": "addr"
 }, {
 "title": "Summary",
 "name": "summary"
 }]
}
```

```
"componentActions": [{
 "name": "infoNext",
 "actionType": "normal",
 "appearance": {
 "label": "Next",
 "style": "primary",
 "belongsTo": "info"
 }
}, {
 "name": "addrBack",
 "actionType": "normal",
 "appearance": {
 "position": "bottom-left",
 "label": "Back",
 "style": "secondary",
 "belongsTo": ["addr"]
 }
}]
```

## 2.9 Portal Modularization II – Steps / Tabs Enhancements

- `current` can be assigned not only a number value as tab/step index but also a string value as the unique tab/step name.
  - Of course, the tab/step with the unique name should be visible (`render` should be true).

URL: <https://.../demo?startStep=dInfo>

```
"global": { "startStep": "$url(startStep); }

{ "current": "$global.attr(startStep);",
 "titles": [
 { "name": "bInfo", "title": "Borrower Information" },
 { "name": "cbInfo", "title": "Co-Borrower Information" },
 { "name": "dInfo", "title": "Documents" },
 { "name": "sInfo", "title": "Summary" }
]
}
```

## 2.9 Portal Modularization III – Cross-Actor Validation

- A save or submit action in an actor can trigger front-end validation in multiple actors before the data is posted to the backend.

```
{
 "name": "actionName",
 "actionType": "submit",
 "validation": ["actor1", "actor2", ...],
 ...
}
```

- A normal action won't trigger front-end validation by default. But it can be configured to do the front-end validation before its handler is triggered.

## 2.9 Portal Modularization IV – Query Default New

- Actor will get and display a new record if it cannot retrieve any existing record from the backend

```
{
 "structureName": "account_detail_view",
 "tableName": "account",
 "whereClause": { "freeForm": "id = ':accountId'" },
 ...
}

"property": {
 "whereClause": { "accountId": "$global.attr(highlightedAccountId); " },
 "structureName": "account_detail_view",
 "queryWithDefaultNew": true
}
```

## 2.9 Material UI

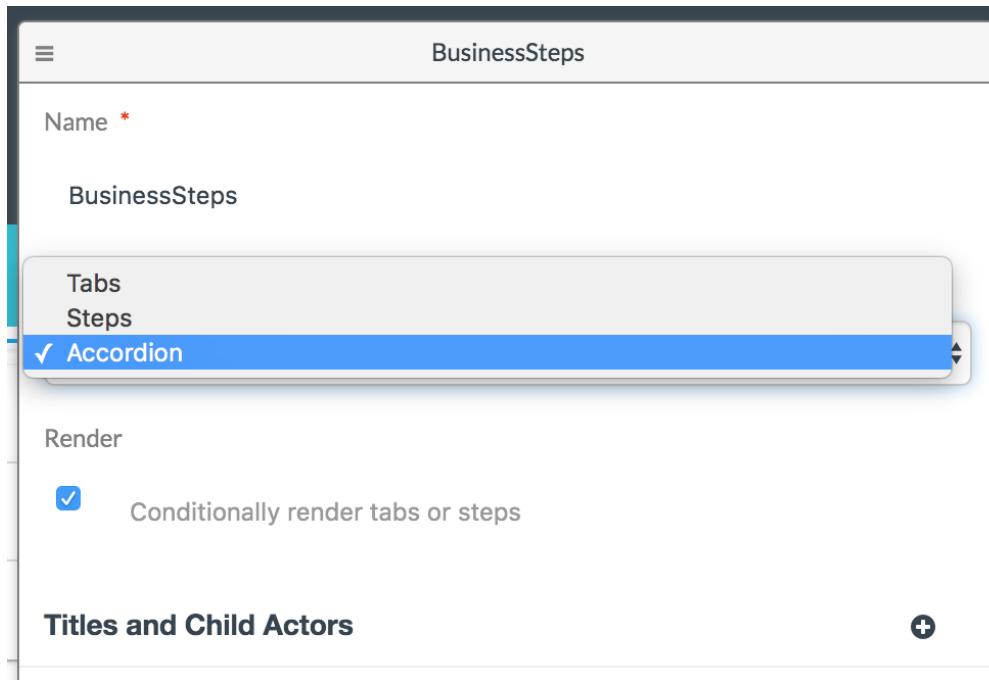
1. Accordion \*
2. Container-level validation
3. Responsive layout grid \*
4. Range field
5. Field alignment \*
6. Queue view \*
7. List view \*
8. Card view \*
9. Pagination
10. Messages \*
11. Group by
12. Collapsible hidden fields \*

The configuration of Material UI and the one of the existing (Mint) UI are not 100% compatible.

The configuration inside the sections whose title ends with \* is MUI specific.

## 2.9 Material UI - Accordion

- Material UI support accordion, besides tabs and steps.



## 2.9 Material I - Accordion

- Accordion expand icon can be customized using  
<https://material.io/tools/icons/>
- Default expand icon is `expand_more`.

```
"appearance": {
 "accordionExpandIcon": "arrow_drop_down_circle",
 "titles": [...]
}
```

## Business Information



### Business Name \*

This is a company name.

### Annual Revenue

\$

### Account Number

### My New PickList

This is the tooltip

## Business Information



### Business Name \*

This is a company name.

### Annual Revenue

\$

### Account Number

### My New PickList

This is the tooltip

### Business Address



### Summary



## 2.9 Material II – Container-Level Validation

- When an **accordion/tabs/steps** includes several editable views/panels, it can be configured to indicate validation errors at the title when the corresponding panel that has the problematic view is collapsed.
  - The accordion must be **sticky** in order to keep the validation errors in those collapsed panels.
  - User can specify suffix icon or prefix icon (using `validationErrorPrefixIcon`). The icon name must be a valid Material UI Icon name

```
"appearance": {
 "validationErrorSuffixIcon": "warning",
 "showValidationError": true,
 "sticky": true,
 "titles": [...]
}
```

Personal Information 



First Name \*

Required

Last Name \*

Required

Social Security Number

Marital Status

Not Provided

Date of Birth

mm/dd/yyyy

Phone Number

Email Address \*

Required

Identification Type

Issuer

Identification Number

Address 



Employment 



Other Income

CANCEL

NEXT

SAVE

## 2.9 Material III – Responsive Layout Grid

- At this point, we only support two grid breakpoints: sm and md
- If not specified, field width is sm=12 (the full width)
- In detail view, panel, and editable queue / card view, users can specify column grid width for each field.
- In card view, users can specify grid width for each card.

```
"appearance": {
 "grid": {
 "field": {
 "AnnualRevenue": { "sm": 6 },
 "Industry": { "sm": 6 },
 "Name": { "sm": 12 }
 }
 } }

"appearance": {
 "grid": {
 "field": {
 "$default": { "sm": 6 }
 } } }

}
}
```

Account Name  
United Oil & Gas Corp.

Industry Annual Revenue  
Energy £ 5470000000

Account Name Account Phone  
United Oil & Gas Corp. (212) 842-5500

Industry Annual Revenue  
Energy £ 5,470,000,000

```
"appearance": {
 "grid": {
 "card": {"sm": 6 }
 } }
}
```

United Oil & Gas Corp.  
£ 5,470,000,000.00

Account Name United Oil & Gas Corp.  
Industry Energy  
Annual Revenue £5,470,000,000.00

Dickenson plc  
£ 500,000,000.00

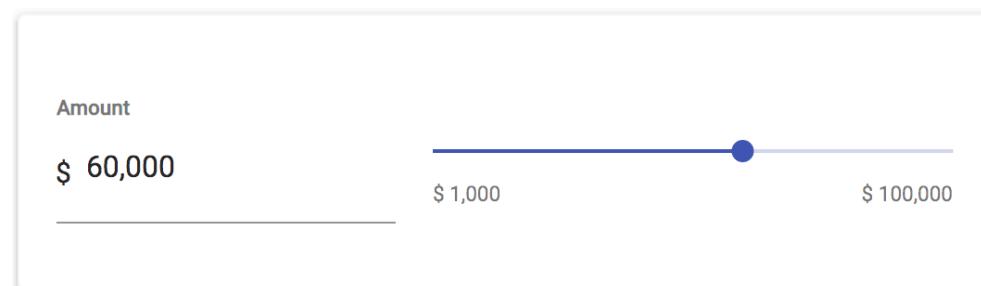
Account Name Dickenson plc  
Industry Consulting  
Annual Revenue £500,000,000.00



## 2.9 Material IV – Range Field

```
{ "fieldName": "...",
 "fieldUi": {
 "fieldType": "range", "label": "Amount",
 "rangeOptions": {
 "dataType": "currency",
 "min": 1000, "max": 100000, "step": 1000,
 "editableRangeValue": true
 },
 "settingsOptions": { "minimumFractionDigits": 0 },
 "format": {
 "options": {
 "numeral": true,
 "numeralThousandsGroupStyle":
 "thousand",
 "numeralDecimalScale": 0
 },
 "prefix": "$"
 }
 }
}
```

In 2.9, **editable** range field **MUST** have format options.



## 2.9 Material V – Field Alignment

- Can be applied to list-view fields and editable (detail-view) fields

```
"appearance": { "fieldAlignment": { "AnnualRevenue": "right" } }
```

£ 7,094,984,523.00		
<input type="checkbox"/>	Account Name	Industry
<input type="checkbox"/>	United Oil & Gas Corp.	Energy
		£ 5,470,000,000.00
<input type="checkbox"/>	Dickenson plc	Consulting
		£ 500,000,000.00
<input type="checkbox"/>	Grand Hotels & Resorts Ltd	Hospitality
		£ 500,000,000.00

Account Name	Account Phone
United Oil & Gas Corp.	(212) 842-5500
Industry	Annual Revenue
Energy	£
	540,021,000

```
"appearance": { "fieldAlignment": { "amount": "right" }, }

{ "fieldName": "amount",
 "fieldUi": {
 "fieldType": "range", "label": "Amount",
 "rangeOptions": { ... },
 "format": { ... } } }
```

Amount

\$

70,000

\$ 1,000

\$ 100,000



## 2.9 Material VI – Queue View

- **Read-only queue view uses** `queuePrimaryText` **and** `queueSecondaryText` **to render**
  - `queuePrimaryText` **and** `queueSecondaryText` are configured in the same way as `title`, `subTitle`, etc., including both the content and the conditional rendering
  - `recordActions` for each queue view entry are recommended to use iconic actions with [Material UI icons](#). (`label` is the icon name.)
- **Read-only queue view without** `queuePrimaryText` **and** `queueSecondaryText` **will be displayed as an empty row.**

<input type="checkbox"/>	United Oil & Gas Corp. Energy – £ 5,470,000,000.00	 
<input type="checkbox"/>	Dickenson plc Consulting – £ 500,000,000.00	 
<input type="checkbox"/>	Grand Hotels & Resorts Ltd Hospitality – £ 500,000,000.00	 

```

"appearance": {
 "queuePrimaryText": "$this.attr(Name);",
 "queueSecondaryText": "$this.attr(Industry); — $this.attr(AnnualRevenue, currency);",
 ...
}

"recordActions": [
 {
 "name": "preview",
 "actionType": "normal",
 "appearance": { "label": "visibility", "style": "iconic" }
 },
 {
 "name": "upload",
 "actionType": "upload",
 "appearance": { "label": "cloud_upload", "style": "iconic" }
 }
]
```

## 2.9 Material VII – List View

- List view can also support inline recordActions, which are displayed in the right-most column

```
"recordActions": [{
 "name": "remove",
 "actionType": "normal",
 "appearance": { "style": "iconic", "label": "remove_circle_outline" }
 ...
}]
```

Income Type	Amount	Action
Others	\$	⊖
Others	\$	⊖
<button>ADD ADDITIONAL INCOME</button>		

## 2.9 Material VIII – Card View

- Card view supports inline recordActions.
- **Read-only** card view uses `cardTitle` and `cardSubtitle` to render
  - `cardTitle` and `cardSubtitle` are configured in the same way as title, subTitle, etc., including both the content and the conditional rendering
- Read-only card view has label-value entries, where label and value are side by side.

## United Oil & Gas Corp.

£ 5,470,000,000.00

Account Name      United Oil & Gas Corp.

Industry              Energy

Annual Revenue      £5,470,000,000.00



## Dickenson plc

£ 500,000,000.00

Account Name      Dickenson plc

Industry              Consulting

Annual Revenue      £500,000,000.00



Edit Account

```
"appearance": {
 "cardTitle": "$this.attr('Name') ;",
 "cardSubtitle": "$this.attr('AnnualRevenue', currency) ;",
 "grid": { "card": { "sm": 6 } }
}
"recordActions": [{
 "name": "...",
 "appearance": { "label": "share", "style": "iconic" }
}, {
 "name": "...",
 "appearance": { "label": "edit", "style": "iconic", "tooltip": "Edit Account" }
}]
```

## 2.9 Material Card View (cont.)

- By default, record actions in each card is at the bottom of the card and right aligned.
- Actions can be positioned to left by specifying the position value.

Express Logistics and Transport 2  
£ 95,000,000.00

Account Name	Express Logistics and Transport 2
Industry	Insurance
Annual Revenue	£95,000,000.00



```
"recordActions": [{
 "name": "foo",
 "appearance": {
 "style": "iconic",
 "label": "share",
 "position }, . . .
}, . . .
```

Express Logistics and Transport 2  
£ 95,000,000.00

Account Name	Express Logistics and Transport 2
Industry	Insurance
Annual Revenue	£95,000,000.00



## 2.9 Material VIII – Pagination

- Pagination in Material UI is displayed at the bottom of each view and center aligned.
- In material UI pagination configuration, link texts must be **material UI icon names**.

<input type="checkbox"/>	United Oil & Gas Corp.	Energy	– £ 5,470,000,000.00		
<input type="checkbox"/>	Dickenson plc	Consulting	– £ 500,000,000.00		
<input type="checkbox"/>	Grand Hotels & Resorts Ltd	Hospitality	– £ 500,000,000.00		

1 - 3 of 22    |< < 1 2 3 4 5 > >|

**Pagination**

Pagination and Grouping cannot be enabled together.

Enable Pagination

Entries per Page \*  Page Links \*

Pagination Text Template \*

`$start; - $end; of $total;`

First Link Text \*

Previous Link Text \*

Next Link Text \*

Last Link Text \*

## 2.9 Material X - Messages

- Loading – global "circular" or local "linear indeterminate"
- Remote animation – global "circular"
  - Including save and submit
- Remote error – global "snackbar"
  - Including save, submit, and upload
- Uploading – "linear determinate" at top or bottom
- Loading error – local message

## 2.9 Messages (cont.)

- Material UI message has simpler configuration options.

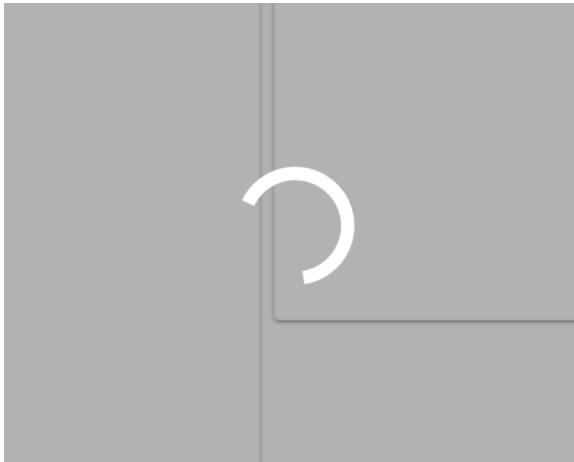
The screenshot shows a configuration interface for 'Messages' in a sidebar. On the left, there is a vertical toolbar with icons for file, person, list, grid, and text, with the text icon being highlighted. The main area is titled 'Messages' and contains several configuration items:

- Animation message when loading data is in progress
- Position \*  
Global
- Animation message when saving data is in progress
- Confirmation message when saving data succeeded
- A** Text \*  
Succeed
- Error message when loading data failed
- Failed to load**
- Error message when saving data failed
- Failed to save**

## 2.9 Messages (cont.) - Loading Message

- By default, loading message is not displayed until the first checkbox is checked.

```
"messagePositions": {
 "loadingAnimation": "global"
}
```

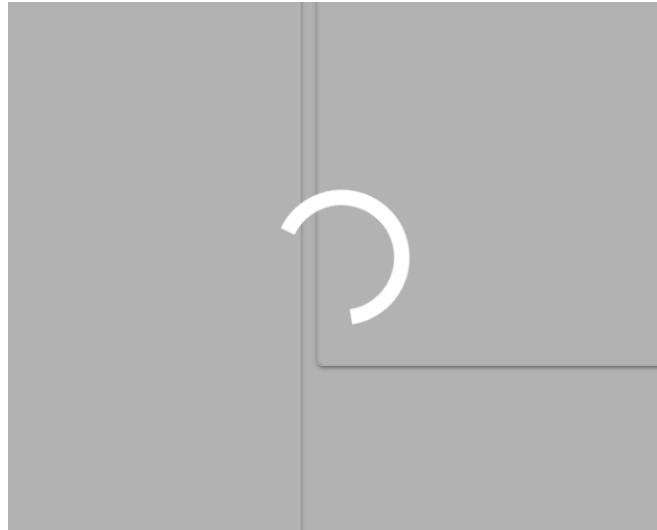


```
"messagePositions": {
 "loadingAnimation": "top"
}
```

Account Name	Industry	Annual Revenue

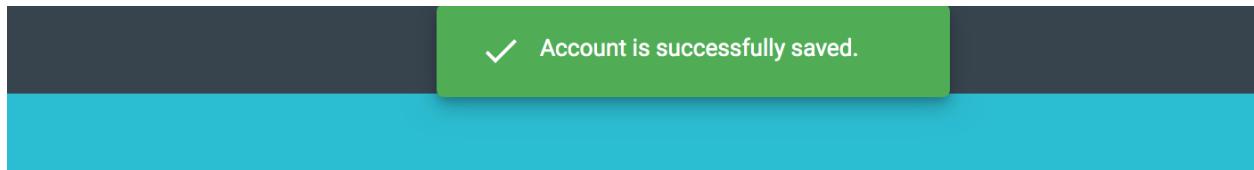
## 2.9 Messages (cont.) - Saving Animation

- By default, saving animation is not displayed until the second box is checked.
- Saving animation is displayed as a global circular progress



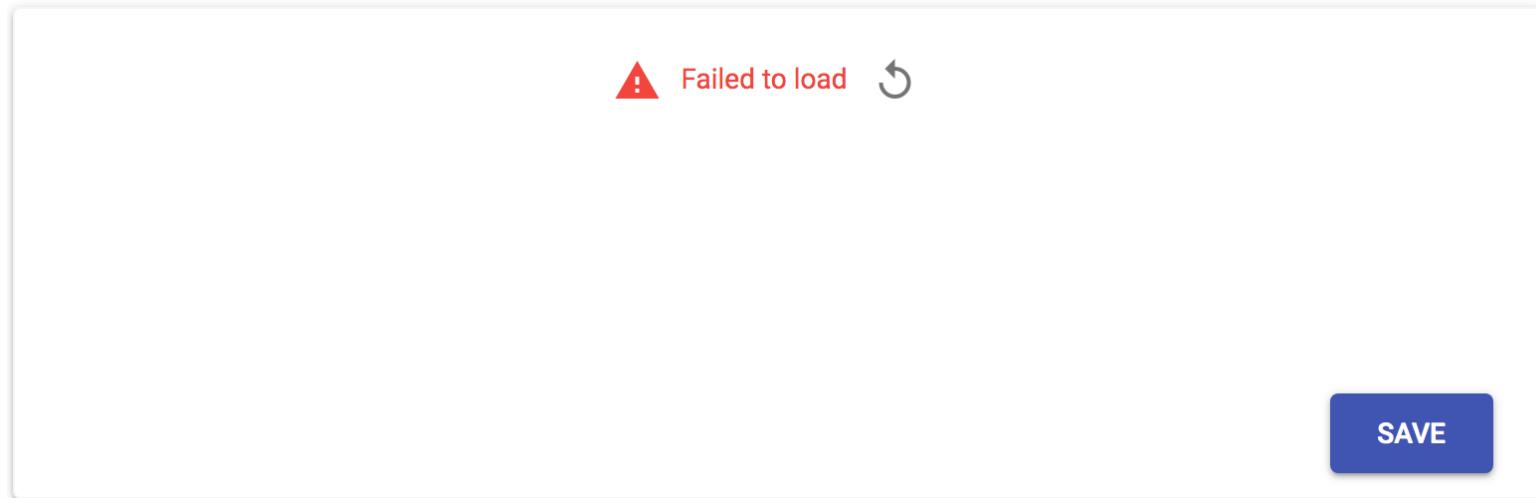
## 2.9 Messages (cont.) - Saving Success

- By default, saving success confirmation is not displayed until the third box is checked.
- Saving success confirmation is displayed as a global "snackbar" in green.



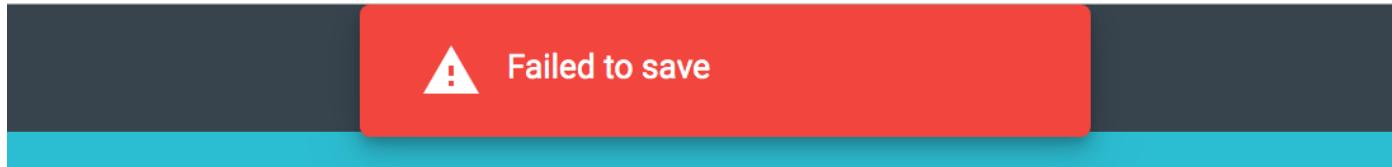
## 2.9 Messages (cont.) - Loading Error

- By default, loading error message is displayed until the fourth checkbox is unchecked.
- Loading error message is displayed at the top of an actor.



## 2.9 Messages (cont.) - Saving Error

- By default, saving error message is displayed until the last checkbox is checked.
- Saving error message is displayed as a global "snackbar" in red.



## 2.9 Messages (cont.) - Remote Submit Animation

- By default, submission animation is displayed the action is configured not to show it
- Submission animation is displayed as a global circular progress

```
"componentActions": [{
 "name": "loanSave",
 "actionType": "submit",
 "appearance": { ... }
 "messages": { "showSubmitAnimation": false }
} ...]
```

## 2.9 Messages (cont.) - Remote Submit Success / Error

- By default, submission success message is not displayed the action is configured to show it
  - Submission success is displayed as a global Snackbar in green.
- By default, submission error is displayed until the action is configured not to show it
  - Submission error is displayed as a global Snackbar in red.

```
"componentActions": [{
 "name": "loanSave",
 "actionType": "submit",
 "appearance": { ... }
 "messages": { "showSubmitError": false, "showSumitSuccess": true }
} ...]
```

## 2.9 Material XI – Group By

- In Material UI, `groupBy` is supported by queue view and card view.
  - Group can be expanded / collapsed if `collapsibleIcons` and `collapsed` are specified.

```
"groupBy": {
 "collapsibleIcons": { "collapsed": "expand_less", "expanded": "expand_more" },
 "collapsed": false,
 "field": "Industry",
 "ascending": false,
 "titles": [{ "title": "$this.attr(Industry);" }]
}
```

## GenePoint 2

£ 30,000,000.00

Account Name GenePoint 2

Industry Biotechnology

Annual Revenue £30,000,000.00



## Energy

### United Oil & Gas, Singapore

£ 0.00

Account Name United Oil & Gas, Singapore

Industry Energy

Annual Revenue £0.00



### United Oil & Gas, UK

£ 0.00

Account Name United Oil & Gas, UK

Industry Energy

Annual Revenue £0.00



## GenePoint 2

£ 30,000,000.00

Account Name GenePoint 2

Industry Biotechnology

Annual Revenue £30,000,000.00



## Energy

## Education

### University of Arizona

£ 0.00

Account Name University of Arizona

Industry Education

Annual Revenue £0.00



## 2.9 Material XII – Collapsible Hidden Fields

- In Material UI, `entryCollapse` is only supported by card view
  - Each record (card) is a card view can be expanded / collapsed.
  - The expanded/collapsed icon is displayed as the rightmost icon with the record actions.

```
"appearance": {
 "entryCollapse": {
 "collapsed": true,
 "collapsibleIcons": {
 "collapsed": "expand_more",
 "expanded": "expand_less"
 },
 "collapsedHiddenFields": ["Name", "Industry"]
 },
 ...
}
```



## Edge Communications

£ 139,000,000.00

Annual Revenue

£139,000,000.00



## Edge Communications

£ 139,000,000.00

Annual Revenue

£139,000,000.00



Account Name    Edge Communications

Industry              Electronics

## 2.10 New Features

1. Make the configuration and the behavior of the picklist and the reference consistent
2. Dependent picklist
3. Dependent reference
4. Reference default value
5. Default value can take variables
6. Material UI: Select buttons and prefix/suffix icons

## 2.10.1 - Consistent Picklist and Reference

- If there is a `defaultValue`, `defaultValue` will be selected by default.
- Else if there is a `prompt`, the `prompt` will be added as the first option and selected.
  - If a picklist is not required, use `prompt` so that an undefined value can be set.
- Otherwise, the first option will be selected.

## 2.10.2 - Dependent Picklist

- Option values in picklist can be conditionally rendered. It works for picklist, radiobuttons, multiValueIndicator, and enhancedPicklist (**Mint only**)
  - Options without `optionRender` appears by default

```
"fieldType": "picklist", "label": "Industry",
"selectOptions": [
 {"optionText": "Agriculture", "optionValue": "Agriculture",
 "optionRender": "$this.attr(Rating) == \"Cold\""
 },
 {"optionText": "Construction", "optionValue": "Construction",
 "optionRender": "$this.attr(Rating) == \"Warm\""
 },
 {"optionText": "Other", "optionValue": "Other"
 }
]
```

## 2.10.3 - Dependent Reference

- Reference values can be conditionally fetched by specifying its whereClause.

```
{
 "fieldName": "Account.Industry",
 "fieldUi": { "fieldType": "picklist", "selectOptions": [...], ... }
}, {
 "relationType": "reference", "fieldName": "AccountId", "relationName": "Account",
 "relationJson": {
 "tableName": "Account",
 "whereClause": { "freeForm": "industry = ':industry'" },
 ...
 } }
}
```

relationName is the key

```
"property": {
 "referenceWhereClause": {
 "Account": { "industry": "$this.attr(Account.Industry); " }
 },
 "whereClause": { "contactId": "003f400000ekVXXAA2" }
}
```

## 2.10.4 - Reference with Default Value

- Reference field can set the default value
  - `defaultValue` is the visible text instead of the record Id

```
{
 "relationType": "reference",
 "fieldName": "AccountId",
 "relationName": "Account",
 "relationJson": {...}
 "fieldUi": {
 "fieldType": "reference",
 "defaultValue": "Q2 eBanking"
 }
}
```

## 2.10.5 - Default Value of Field with \$variables

```
"fields": [{
 "fieldName": "FirstName",
 "fieldUi": {
 "fieldType": "string", "label": "First Name",
 "defaultValue": "$global.attr(userInfo).attr(firstName);"
 }
, {
 "fieldName": "LastName",
 "fieldUi": {
 "fieldType": "string", "label": "Last Name",
 "defaultValue": "$global.attr(userInfo).attr(lastName);"
 }
} ...]
```

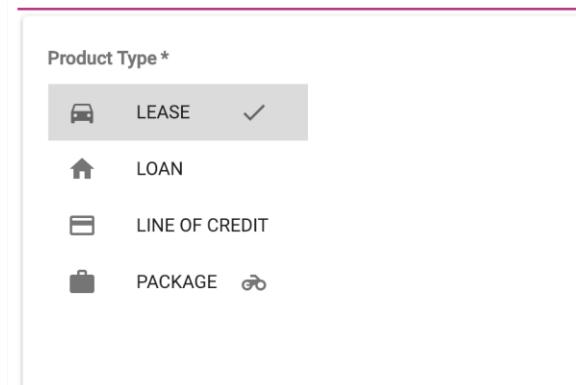
## 2.10.5 - Default Value (cont.)

- Use default value including `$variable` **with care**
  - Should not create a dependency cycle
  - If you are not sure whether the dependent `$variable` is available before the evaluation, do not use this feature.
    - For example, you should not have `$this.attr(...)` as the default value since when a record data is read from the backend, the value order is not guaranteed.
  - **Recommended best practice:** only use `$global` and `$currentStage` in the default value expression.

## 2.10.6 - Material UI: Select Buttons

Picklist can be rendered as Select buttons with either prefix or suffix or both prefix and suffix icons, it supports all the Material icons. It also supports selectedSuffix and selectedPrefix options for the select buttons

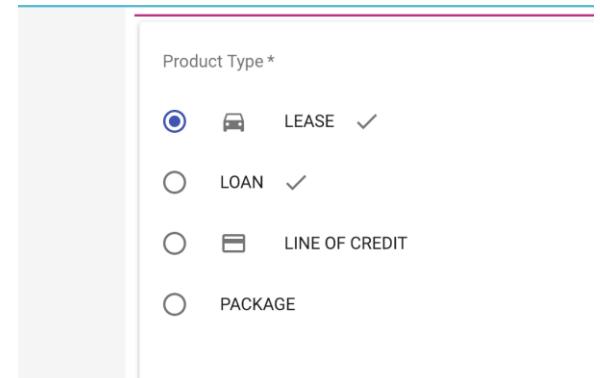
```
"fieldType": "selectbuttons", "label": "Account Type",
"selectOptions": [{
 "id": "Prospect", "active": true, "defaultOption": false,
 "optionText": "Prospect", "optionValue": "Prospect",
 "optionPrefix": "home",
 "optionSuffix": "done"
},
 "selectedPrefix": "",
 "selectedSuffix": "
```



# Prefix and Suffix icons for Picklist & Radio buttons

Material icons can be added as prefix or suffix or both to radio buttons and Picklist as options

```
"fieldType": "selectbuttons", "label": "Account Type",
"selectOptions": [
 {
 "id": "Prospect", "active": true, "defaultOption": false,
 "optionText": "Prospect", "optionValue": "Prospect",
 "optionPrefix": "home",
 "optionSuffix": "done"
 },
}
```



## 2.10 Accessibility

- Disclosure
- Dialog
- Images
- Links
- Range Slider
- Page-include
- Static Content
- Actions
- List-view
- Queue-view (in progress)
- Card-view (in progress)
- Detail-view (in progress)

## 2.10 Accessibility – Disclosure

- Disclosures fully functional with keyboard
  - Disclosure description is tabbable and read by screen readers
  - Disclosure checkbox comes with aria-label="Disclosure acknowledgment"
- No configuration change

## 2.10 Accessibility – Dialog

- Dialogs fully functional with keyboard
  - Dialog title is tabbable and read by screen readers
  - Dialog content is tabbable and read by screen readers
  - Dialog confirm and cancel buttons are tabbable and have aria-labels "Confirm button" and "Cancel button" respectively
- No configuration change

## 2.10 Accessibility – Images

- Images fields can now be fully configured in builder
  - Images tabbable and alt-text read by screen reader
  - Alt-text for images configurable in builder
  - Image src configurable in builder

Accessibility

Image Alt Text \*

## 2.10 Accessibility – Images (cont)

Example JSON

```
{
 "id": "a09a252f-7e24-4dd7-9ea6-cc2fec53564b",
 "fieldName": "AccountNumber",
 "fieldUi": {
 "fieldType": "image",
 "label": "Account Number",
 "defaultValue": "https://picsum.photos/id/989/200/200",
 "imageAltText": "Lorem ipsum picture"
 }
}
```

## 2.10 Accessibility – Links

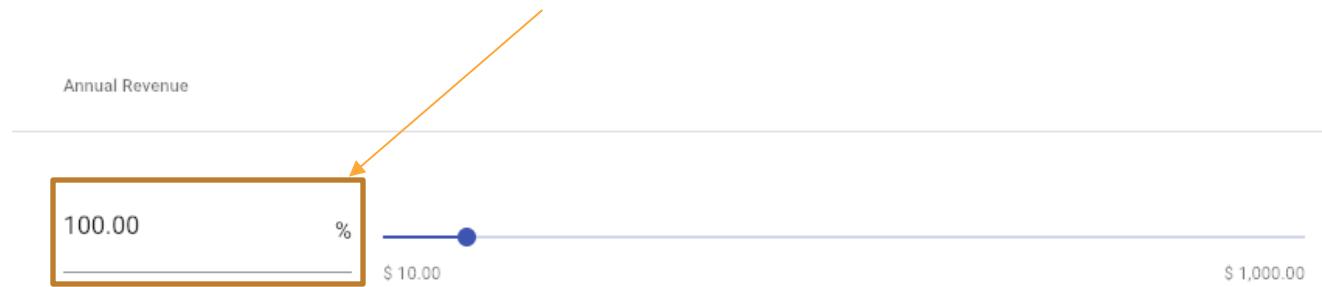
- Link fields can be fully configured in builder
  - Links tabbable
  - Link aria-labels added to builder
    - Supports \$ variables (E.g. \$global.attr(websiteName);)
    - Read by screen-reader over actual link text
  - Works with relation fields (I.e. "clcommon\_\_Navigation\_Structure\_\_r.Name")

## 2.10 Accessibility – Links (cont)

```
{
 "id": "49062480-f0ea-4e03-acfe-1ec31eb354d1",
 "fieldName": "Name",
 "fieldUi": {
 "fieldType": "link",
 "label": "Account Name",
 "defaultValue": "Link to Google",
 "linkTemplate": "$global.attr(google);",
 "ariaLabel": "Google link"
 }
}
```

## 2.10 Accessibility – Range Slider

- Range Slider accessibility limitations due to material-ui v3
  - Range Slider TextField not tabbable



## 2.10 Accessibility – Page-include

- Page-include fully configurable in builder
  - ariaLabel attribute to content property
  - aria-label tag added to iframe

## 2.10 Accessibility – Static Content

- Static Content fully configurable in builder
  - ariaLabel attribute to content property (same attribute used in page-include)
  - aria-label tag added to div containing static content

## 2.10 Accessibility – Page-include & Static Content

Example JSON:

```
Property: {
 "ariaLabel": "Page-include or static content aria label"
}
```

## 2.10 Accessibility – Actions

- New attribute ariaLabel added to action configuration
  - "ariaLabel": "Insert aria label here"
- Screen reader will read aria-label over label text
- Menu buttons are identified as popup buttons by screen readers
- Action tooltip will be read by screen reader in addition to aria-label

## 2.10 Accessibility – Actions

```
{
 "name": "next1",
 "actionType": "upload",
 "property": {},
 "appearance": {
 "style": "primary",
 "label": "Next",
 "position": "bottom-right"
},
 "successHandler": {},
 "failureHandler": {},
 "pendingHandler": {},
 "handler": {},

```

## 2.10 Accessibility – Components

```
Appearance: {
 "accessibility": {
 "ariaLabel": "aria label for component actors"
 "headerField": "AnnualRevenue"
 }
}
```

Note: headerField used only by List-view

## 2.10 Accessibility – List view

- List-view table is tabbable
- List-view aria-label configurable in builder
- Table row header configurable in appearance JSON, if none specified, first field becomes row header
- When entries are selectable, checkbox will reference headerField on that row
- Select all checkbox comes with dynamic aria-labels denoting if all checkboxes are checked
- Pagination buttons comes with hard-coded aria-labels
- Sortable fields have dynamic aria-labels defining direction of sort

## 2.10 Accessibility – Queue view

- Queue-view list is tabbable
- Queue-view aria-label configurable in builder
- Selectable checkboxes labelled by Primary & Secondary text
- Pagination buttons comes with hard-coded aria-labels
- Grouping introduces aria-label linking using ids to give meaning groups for screen readers. Groups are referenced by their group titles
- Collapsible groups come with dynamic aria-label to denote whether group is collapsed or expanded
- NOTE: Sorting not accessible

## 2.10 Accessibility – Card view

- Card-view grid is tabbable
- Card-view aria-label configurable in builder
- Selectable checkboxes has aria-labels referencing cardTitle & cardSubtitle
- Pagination buttons comes with hard-coded aria-labels
- Grouping introduces accessible grouping. This allows screen readers to know what cards are a part of which group
- Collapsible groups come with dynamic aria-label to denote whether group is collapsed or expanded. Group level and record level
- NOTE: Sorting not accessible

## 2.10 Accessibility – Detail view

- Detail-views are tabbable
- Detail-view aria-label configurable in builder

## 2.10 Accessibility – Form fields

- Form fields have unique ids formed by their recordId and fieldId
  - This allows for quicker accessibility configuration relating to List-view, queue-view, etc
- Form fields have aria-labels configured in builder
  - Aria-labels are excluded for TextFields since TextFields are accessible by default through material ui
- Note: Picklist, multipicklist, and range slider are excluded from above due to limitations of material ui v3
- Reference JSON configuration from Link accessibility

## 2.10 Builder Loader Enhancement

- When executing `builder-loader.js` from the command line, the first two arguments are username and password to a SFDC org
- The rest three named arguments are optional
  - Org's security token: `token=n1UdUj31razvjVRmuEsmHCYt3`
  - Sandbox or not: `sandbox=true` (If `sandbox` is true, the login URL is <https://test.salesforce.com>)
  - namespace prefix: `prefix=testApplication` (`_` added at the end automatically if not provided)
    - If prefix is not specified, the default namespace prefix is `clcommon_`
    - If prefix is set as `prefix=`, the namespace prefix is an empty string.

## 2.10 Builder Loader Enhancement (cont.)

- For example
  - \$ node builder-loader.js min@portal.playground merc1237 prefix=testApplication token=n1UdUj31razvjVRmuEsmHCYt3
  - \$ node builder-loader.js min@portal.playground merc1237 **prefix=sandbox=true** token=n1UdUj31razvjVRmuEsmHCYt3

## 2.10 Mui Branding

- 2.9 upgrade: clcommon\_\_css\_configuration\_\_c
  - clcommon\_\_content\_\_c
    - Inside layoutMapping > ClsAppLayoutOne > styleSpecification, change path **and** cssSelector **to** category
    - Refer to the latest applayout1style.json
  - clcommon\_\_branding\_\_c
    - Only keep theme **and** fontFamilies
    - Refer to the latest style.mui.json

## 2.10 Mui Branding

- Style converter (Mui Only)
  - node builder-converter-29-to-210.js *username password token=token* **pageId=pageId**
  - For example
    - node builder-converter-29-to-210.js min.wu-tfb-dev@q2ebanking.com merc1235 token=XWu8vjvKR7HH13J05cAHszis3 **pageId=BorrowerPortal**

## 3.2 stylingClassNames

- Third-party class names can be added to different UI widgets through appearance

```
"appearance": {
 ...
 "stylingclassNames "actor": ...
 "record": ...
 "section": ...
 "field": ...
 "title": ...
 "subtitle": ...
 "text-before": ...
 "text-after": ...
 "blank-message": ...
 "label": ...
 "actions "default": ...,
 "actionName1": ...
 "actionName2": ...
 }
 }
}
```

## 3.2 Field Validation

```
{ "fieldName": "annualRevenue",
 "fieldUi": {
 "fieldType": "currency",
 "label": "Annual business revenue",
 "required": true,
 "validation": [
 {
 "rule": "$this.attr(annualRevenue) >= 50000",
 "message": "Minimum of $50,000 annually"
 },
 "messages": { "requiredErrorText": "Minimum of $50,000 annually" }
],
 "validation": [
 {
 "rule": "$this.attr(borrowedAmount) <= $global.attr(offeredAmount);",
 "message": "You can borrow up to $global.attr(offeredAmount, currency);."
 },
]
}
```

## 3.2 Field Validation with Custom JS Code

```
"fields": [{
 "fieldName": "Birthdate",
 "fieldUi": {
 "fieldType": "date",
 "validation": [{
 "rule": "$ext.confirmAge",
 "arguments": ["$this.attr(Birthdate);", 21],
 "message": "You have to be 21 years old."
 }],
 ... }]
```

## 3.2 Validate related field

- Through assignment: changing field A assigns a value to field B.
  - If field B has some validation error, and changing A assigns a valid value to B, B's validation error should be removed.
  - However, if changing A assigns some invalid value to B, field B won't automatically display the validation error. The error will be displayed if field B is on focus and then on blur or a button is clicked to trigger the field validation.
- Through cross-field validation: changing field A affects the validation of field B

## 3.2 Cross-Field Validation

- A field (`minAmount`) may have a `relatedFields` entry, which is a list of field names (e.g, `[loanAmount]`) in the current actor.
  - When the value of the field (`minAmount`) is updated and validated, the related fields are validated automatically.
- A field (`loanAmount`) may have a `validation` entry, which has
  - `condition`: if the condition is evaluated true, check the validation rule.
  - `rule`: if the rule is evaluated false, the message is displayed.
  - `message`

## 3.2 Cross-Field Validation

```
"fields": [
 { "id": "74a8",
 "fieldName": "minAmount",
 "fieldUi": {
 "fieldType": "currency",
 "label": "Minimum Amount",
 "required": true,
 "relatedFields": ["loanAmount"],
 "format": {
 "options": {
 "numeral": true,
 "numeralThousandsGroupStyle": "thousand"
 } } },
 { "id": "ea6f",
 "fieldName": "loanAmount",
 "fieldUi": {
 "fieldType": "currency",
 "label": "Loan Amount",
 "required": true,
 "validation": [
 {
 "rule": "$this.attr(loanAmount) ; >= $this.attr(minAmount) ;",
 "condition": "$this.attr(minAmount) ; != undefined and $this.attr(loanAmount) ; != undefined",
 "message": "Loan amount cannot be less than $this.attr(minAmount, currency) ;."
 }],
 "format": {
 "options": {
 "numeral": true,
 "numeralThousandsGroupStyle": "thousand"
 } } }]}
```

## 3.2 Cross-Field Validation

```
{
 "id": "4782",
 "fieldName": "newPassword",
 "fieldUi": {
 "fieldType": "string",
 "label": "Password",
 "required": true,
 "secure": true,
 "relatedFields": [
 "confirmedPassword"
],
 "messages": {
 "requiredErrorText": "Password is
required."
 }
 }
}

{
 "id": "8f43",
 "fieldName": "confirmedPassword",
 "fieldUi": {
 "fieldType": "string",
 "label": "Confirmed Password",
 "required": true,
 "secure": true,
 "validation": [{
 "rule": "$this.attr(newPassword) ==
>this.attr(confirmedPassword);",
 "condition": "$this.attr(newPassword) !=
undefined and $this.attr(confirmedPassword) !=
undefined",
 "message": "Password and confirmed
password are not same."
 }],
 "messages": {
 "requiredErrorText": "Confirmed password is
required."
 } }]
}
```

## 3.2 Assignment with Custom JS Code

```
"generalHandlers": {
 "fieldChange": {
 "genesis_Residual_Amount_c, genesis_Payment_Frequency_c": {
 "assignments": [{
 "field": "$this.attr(genesis_Interest_Rate_c);",
 "value": "$ext.calculateInterestRate",
 "arguments": ["$this.attr(genesis_Residual_Amount_c);",
"$this.attr(genesis_Payment_Frequency_c);"]
 }, {
 "field": "$this.attr(genesis_Payment_Amount_Per_Period_c);",
 "value": "$ext.calculateMonthlyPayment",
 "arguments": ["$this.attr(genesis_Residual_Amount_c);",
"$this.attr(genesis_Payment_Frequency_c);"]
 }]
 } } }
```

## 3.2 Assignment with Custom JS Code

```
<script type="text/javascript">
 const portalext = {};
 portalext.calculateInterestRate = (amount, term) => { ... };
 portalext.calculateMonthlyPayment = (amount, term) => { ... };
</script>
```

## 3.2 Performance Check

Edit Actor: NavEntries

Delete Actor | Apply | Cancel

Property Appearance Disclosures Actions General Handlers Performance

Enable Performance

Enable Performance

Performance

```
1 < [
2 "component": [
3 "data",
4 "highlighted",
5 "expandedTreeItems"
6]
7]
```

"performanceCheck": {  
 "component": [ "data", "highlighted", ...  
 }  
}

## 3.2 Performance Check

- Performance check can happen at the component level
  - "data": re-render the component only if the data is different, or
  - "highlighted": only re-render the component only if the highlighted record is different, or
  - ...
- Performance check can happen at the entry level (future work)

## 3.2 Action Loading

```
[{
 "name": "...", "actionType": "submit",
 "appearance": {
 "label": "Next",
 "style": "primary",
 "loading": "$global.attr(actionLoading); == true"
 },
 "successHandler": {
 "assignments": [{
 "field": "$global.attr(actionLoading);",
 "value": false
 }]
 },
 "handler": {
 "assignments": [{
 "field": "$global.attr(actionLoading);",
 "value": true
 }]
 } }]
```

Only used by Amex for now

## 3.2 Action Loading (cont.)

- Record action loading (cannot use global attribute) Only used by Amex for now

```
{ "name": "...", "actionType": "submit",
 "appearance": {
 "loading": "$this.attr(recordActionLoading); == true",
 "style": "tertiary iconic", "icon": "dls-icon-share"
 },
 "successHandler": {
 "assignments": [{ "field": "$this.attr(recordActionLoading);", "value": false }]
 },
 "failureHandler": {
 "assignments": [{ "field": "$this.attr(recordActionLoading);", "value": false }]
 },
 "handler": {
 "assignments": [{ "field": "$this.attr(recordActionLoading);", "value": true }] }
```

## 3.2 Updatable Field (Ext only)

- Using Ext renderer, the portal framework should rarely update the field value. The field value should be updated by the JS layer.
- However, there are use cases that changing one field needs to update the value of another field through the `fieldChange` handler. In this case, the field should be marked as `extUpdatable` in the actor's appearance

```
"appearance": {
 "extUpdatable": [...]
}
```

## 3.2 Templates

```
"appearance: {
 ...
 "templates": {
 "loading": "<div role=\"progressbar\" aria-valuemin=\"0\" aria-
 valuemax=\"100\" class=\"progress-circle progress-indeterminate progress-
 lg\"></div>"
 "actorError": "<div class=\"pad-responsive alert dls-red\"
 role=\"alert\"><span class=\"icon dls-icon-warning-filled margin-1-
 r\">$message;</div><hr/>"
 "uploading": "<div role=\"progressbar\" aria-valuemin=\"0\" aria-
 valuemax=\"100\" aria-valuenow=\"$progress;\" class=\"progress-bar\"><div
 class=\"progress-track count-width\"></div></div>"
 }
}
```

\$message; and \$progress; are the placeholder.

Available template keys are loading, actorError, recordError, uploadActorError, uploadRecordError, uploading

## 3.2 Button Enhancement (**Amex only right now**)

- Add the icon attribute for the icon text in the button appearance so that a button can have both an icon and a label.
- iconic style can be combined with primary, secondary, and tertiary

```
"componentActions": [{
 "name": "foo",
 ...
 "appearance": {
 "style": "secondary iconic",
 "label": "Good!",
 "icon": "dls-icon-thumbs-up"
 }, ...
}
```

## 3.2 Extension

Portal framework can be configured to trigger external JS code for certain use cases:

1. Validation
2. Assignment
3. Execution

## 3.2 Extension - Validation

- Use case: validate the customer at least 21 years old based on his/her date of birth

```
{ "fieldName": "Birthdate",
 "fieldUi": {
 "fieldType": "date",
 "label": "Date of Birth",
 "required": true,
 "regex": "^(12)\d{3}-(0[1-9]|1[0-2])-(0[1-9]|12)\d{3}[01])$",
 "messages": { "regexErrorText": "Enter a valid date of birth." },
 "validation": [
 {
 "rule": "$ext.confirmAge",
 "arguments": ["$this.attr(Birthdate);", 21],
 "message": "User has to be 21 years old"
 }
]
 }
}
```

## 3.2 Extension – Validation (cont.)

- Use case: validate the customer at least 21 years old based on his/her date of birth
  - The external JS function return a Boolean value for validation pass or fail

```
(portalext => {
 portalext.confirmAge = (dateValue, minAge) => {
 if (moment) {
 const years = moment().diff(dateValue, "years", false);
 return years >= minAge;
 } else {
 return false;
 }
 };
})((window.portalext = window.portalext || {}));
```

## 3.2 Extension - Assignment

- Use case: loan calculation – when the residual amount or term changes, update interest rate and monthly payment in real time.

You are approved for a loan of up to **\$10,000.00**. Please Verify the amount you need, the term length, and monthly payment to proceed.

Residual Amount

\$ 5,000



\$5,000

\$10,000

Term

24

Interest Rate

4.50

Monthly Payment

% \$ 1,875.90

## 3.2 Extension - Assignment

- Use case: loan calculation – when the residual amount or term changes, update interest rate and monthly payment in real time.

```
"generalHandlers": { "fieldChange": {
 "genesis_Residual_Amount_c, genesis_Payment_Frequency_c": {
 "assignments": [{
 "field": "$this.attr(genesis_Interest_Rate_c);",
 "value": "$ext.calculateInterestRate",
 "arguments": ["$this.attr(genesis_Residual_Amount_c);",
 "$this.attr(genesis_Payment_Frequency_c);"]
 }, {
 "field": "$this.attr(genesis_Payment_Amount_Per_Period_c);",
 "value": "$ext.calculateMonthlyPayment",
 "arguments": ["$this.attr(genesis_Residual_Amount_c);",
 "$this.attr(genesis_Payment_Frequency_c);"]
 }] ...
 } } }
```

## 3.2 Extension - Assignment

- Use case: loan calculation – when the residual amount or term changes, update interest rate and monthly payment in real time.
  - The external JS function return the value that can be assigned to another field in real time.

```
portalext.calculateMonthlyPayment = (amount, term) => {
 const rate = calcInterestRate(amount, term);
 let calTerm = term/12;
 if (rate == 0) { return 0; }
 term = Number.parseInt(term);
 if (Number.isNaN(term)) { return 0; }
 let perRate = rate/12;
 return (amount * (perRate * (1+perRate) ** term)) / ((1+perRate) ** term - 1);
};
```

## 3.2 Extension – Execution

- Use case: when submission fails, if the form is long, scroll the page to the error message at the top of the form.

```
"componentActions": [{
 "name": "getting_started_save_and_continue",
 "actionType": "submit",
 "handler": {...}
 "remoteActionClassname": ..., "remoteActionArguments": {...},
 "failureHandler": {
 "executions": [{ "name": "$ext.scrollToError" }]
 },
}, ...

((portalext, $) => {
 portalext.scrollToError = () => {
 $(".error-template")[0].scrollIntoView();
 };
})((window.portalext = window.portalext || {}), $jQuery);
```

## 3.2 Backend Delay API

- Class name: PortalActions
- Method name: mockResponse(Integer i, String status)
- Parameters:
  - Integer i: number of seconds to wait (0-10)
  - String status: String to denote type of response
    - "SUCCESS"
    - "PENDING"
    - "ERROR"
- Example call: PortalActions.mockResponse(4, "PENDING");
- Expected response: Response object with response status set to "PENDING".

## 3.2 Mui Variant

- Mui textfields and buttons support Material-UI built-in variant.

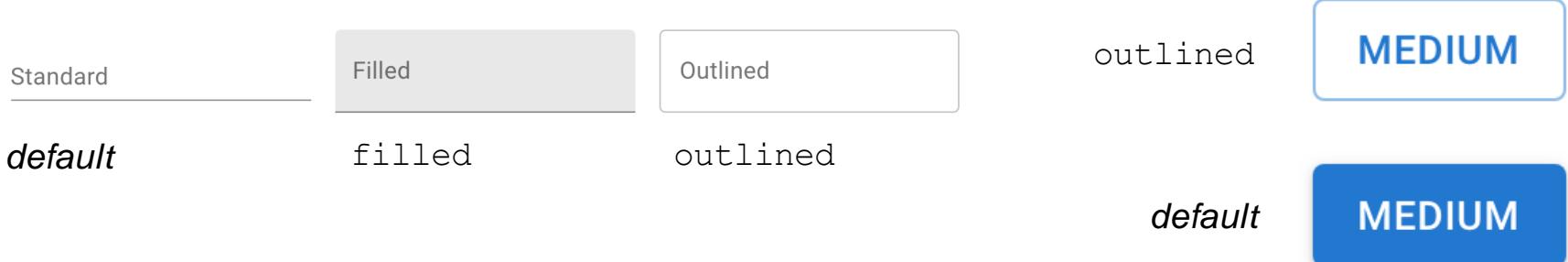
```
"settings": {
 "variant": {
 "TextField": "outlined",
 "Button": "outlined"
 }
 ...
}
```

<https://material-ui.com/components/text-fields/>

<https://material-ui.com/components/buttons/>

text

**MEDIUM**



## 3.2 Mui Stepper Enhancement

- When there are many steps in an Mui Stepper, you can configure the number of visible steps at a time, which is pretty useful for in the mobile view.

```
"appearance": {
 "numberOfVisibleTitles": 3,
 ...
}
```

1 ————— 2 ————— 3 —————

Request      Purpose      Your Credit

✓ ————— 2 ————— 3 —————

Request      Purpose      Your Credit

✓ ————— 3 ————— 4 —————

Purpose      Your Credit      Your Income

BACK

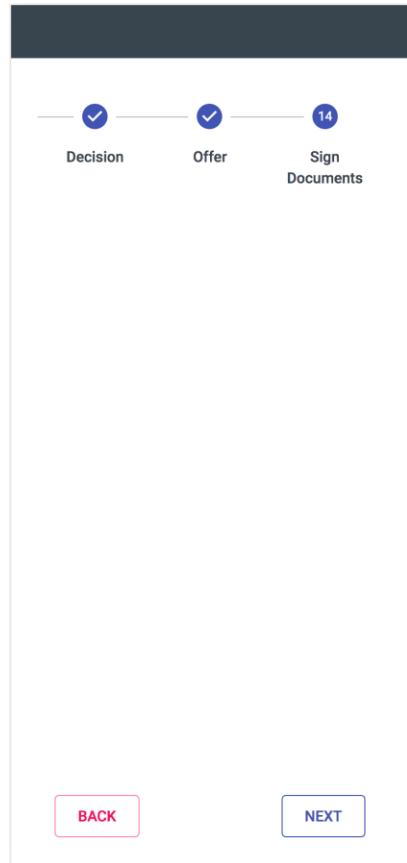
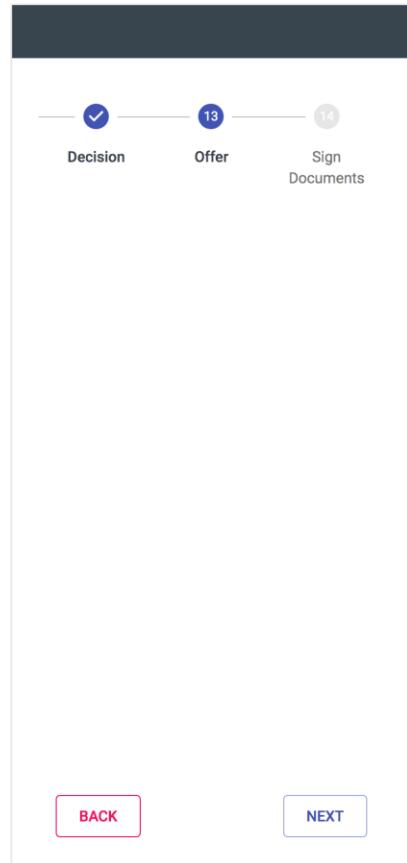
NEXT

BACK

NEXT

BACK

NEXT



## 3.2 Save and Continue - Use Cases

- A user is in an application flow, and he decides to drop and come back to the application later.
- Through a link in the notification email, where the id of his application is part of the URL. The id can be the application id if an application has been created, or an external id otherwise

For example: <https://c.na174.visual.force.com/apex/ConsumerDemo#a8E6g000000TucsEAC>

- Through a dashboard where the user can continue his pending applications.

Application ID	>
APP-0000001131	>
APP-0000001130	>
APP-0000001129	>
APP-0000001128	>

## 3.2 Save and Continue - State Transfer

- Switch stage
  - State name is the stage name that the user just entered
- Switch step/tab/accordion (container)
  - State name is a comma-separated string.
  - The first part is the current stage.
  - The following parts will be the name (or title if there is no name) of the current step/tab/accordion of all the containers in the current stage, from the ancestor to the descendant in case there is a stepper inside a stepper.
- For example: ApplicationTabs,personal

## 3.2 Save and Continue - State Data

- Data to be remembered and get retrieved to continue an application flow.
  - Global data
  - The name of the current stage
  - Prerequisite of the current stage
  - The current index of all the containers in the current stage
- Need to enforce the good practice by decoupling actors

```
{
 "appName": "consumer_demo",
 "channel": "desktop", // "channel": "phone" can be detected only when portal.js (not builder.js) is used.
 "stageName": "ApplicationTabs",
 "stagePrerequisite": {},
 "global": {
 "loanAmount": 20000,
 "applicationId": "a8E6g000000TucsEAC",
 "creditWorth": {
 "fields": {
 "genesis__CL_Product__c": "a0Y6g000001Dj9REAS",
 "genesis__CL_Purpose__r": {
 "Id": "a0Z6g0000017Vd7EAE",
 "Name": "Debt Consolidation"
 },
 "genesis__Term__c": 40,
 "genesis__CL_Product__r": {
 "Id": "a0Y6g000001Dj9REAS",
 "clcommon_Product_Name__c": "Unsecured Consumer Loan",
 "RecordTypeId": "0126g00000181tlAAA"
 },
 }
 },
 "firstName": "John"
 },
 "containerCurrent": {
 "loanApplicationTabs": 2
 }
}
```

## 3.2 Save and Continue - Configuration

```
"settings": {
 "saveContinue": {
 "enable": true,
 "id": "$global.attr(applicationId);", // This is the link Id.
 "excludedStages": ["ChooseLoan", "Home"]
 }
}
```

- There are two ids associated with the save and continue session.
  - The external Id is always the hash value in the URL.
    - <https://...../...#externalId>
  - The link Id is specified in "settings" and **has to be a global attribute** (`$global.attr(...);`)

## 3.2 Save and Continue Configuration

- Action handler to resume the `saveContinue` state

```
"handler": {
 "executions": [{
 "name": "saveContinueResume",
 "value": ["$global.attr(applicationId);"] // The link Id or external Id to retrieve the state
 }]
}
```

## 3.2 Save and Continue – Excluded Stage

- Enter a stage that is excluded from save and continue
  - The URL hash is cleared: <https://c.na174.visual.force.com/apex/ConsumerDemo#>
  - **The link id (e.g., "\$global.attr(applicationId);") will be reset automatically.**

## 3.2 File Upload Accept

- Upload action can specify accepted file types
  - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file#attr-accept>
  - Unique file type specifiers ([https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file#Unique\\_file\\_type\\_specifiers](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file#Unique_file_type_specifiers))
    - A valid case-insensitive filename extension, starting with a period (".") character. For example: .jpg, .pdf, or .doc.
    - A valid MIME type string, with no extensions.
    - audio/\*, video/\*, image/\*

## 3.2 File Upload Accept

```
"componentActions": [{
 "name": "upload",
 "actionType": "upload",
 "property": {
 "multiple": true,
 "accept },
 ...
}]
```

## 3.3 View Template

- Both read-only queue view and read-only card view support view template.

```
"appearance": {
 "templates": {
 "record": "<div class=\"name heading-1\">$this.attr(Name);</div><div
class=\"revenue\">$this.attr(AnnualRevenue, currency);</div><div
class=\"date\">expires on $this.attr(SLAExpirationDate_c, date,
'MM/DD/Y');</div>"
 },
 ...
}
```

# 3.3 View Template with Conditions

- Record template supports conditions

```
"appearance": {
 "templates": {
 "record": [{
 "template": "<div class=\"flex flex-justify-
between\">$this.attr(Name);$this.attr(BodyLength, number,
|KB|');</div>",
 "condition": "$this.attr(BodyLength) < 1000000"
 }, {
 "template": "<div><div class=\"flex flex-justify-
between\">$this.attr(Name);$this.attr(BodyLength, number,
|KB|');</div><div>This document is
missing some information.</div></div>",
 "condition": "$this.attr(BodyLength) >= 1000000"
 }]
 } }
```

## 3.3 Number Format

- We can add number format to portal variables that are embedded in string to improve the flexibility of the date/time representation.
  - Works for the field type (the second argument) of number and currency
  - Format options: ' |MKB| ' with either uppercase or lowercase
- Expected behavior (can be enhanced later)
  - If M/m is set and K/k is set: 16 MB, 357 KB, < 1 KB
  - If M/m is set and K/k is not set: 16 MB, 1 < MB

```
"templates": {
 "record": "<div class=\"flex flex-justify-
between\">$this.attr(Name);$this.attr
(BodyLength, number, ' |KB| ')</div>"
}
```

## 3.3 Number Format

```
{
 "fieldName": "BodyLength",
 "fieldUi": {
 "fieldType": "number",
 "label": "File Size",
 "ariaLabel": "File Size",
 "settingsOptions": { "fieldFormat": " | KB |" }
 }
}
```

File Name	File Size
powered_by_google_on_white_hdpi.png	5 KB
powered_by_google_on_white.png	2 KB
powered_by_google_on_non_white.png	1 KB

## 3.3 Moment Format

- We can add `momentFormat` to portal variables that are embedded in string to improve the flexibility of the date/time representation.
  - Works for the field type (the second argument) of date, time, and datetime.
  - We support all the display format string combinations from Moment.js  
<https://momentjs.com/docs/#/displaying/>

```
"appearance": {
 "templates": {
 "record": "<div class=\"name heading-1\">$this.attr(Name);</div><div
 class=\"revenue\">$this.attr(AnnualRevenue, currency);</div><div
 class=\"date\">expires on $this.attr(SLAExpirationDate_c, date, 'MM/DD/Y | -
 2y')</div>"
 },
 ...
}
```

## 3.3 Moment Format

- The momentFormat has two parts, separated by |
  - The first part is the date and time format, and
  - The second part is the manipulation string.
  - The second part is optional if there is no |.

```
"appearance": {
 "textBefore": "Please select one of the following to upload for
 $this.attr(submissionDate, date, 'MMMM YYYY | -2y -6M'); to
 $this.attr(submissionDate, date, 'MMMM YYYY | -2y -1M');:",
}
```

## 3.3 Text Format

- With two text types: phone and masked
  - The template string has at most 3 parts.
  - The first part is the placeholder character.
  - The optional second part is the mask character.
  - The past part is the format string.

```
$this.attr(phoneNumber, phone, 'x|\\(xxx\\) xxx-xxxx');
=> (972) 555-1212
```

```
$this.attr(ssn, masked, 'x|*|***-**-xxxx');
=> ***-**-6789
```

## 3.3 \$date variable

- To display date information based on the current date and time
- Can be used as a string value (e.g., for query)
- `$date ( 'format | manipulation' ) ;`
  - Current date displayed in a given format (  
<https://momentjs.com/docs/#/displaying/>)
  - y, M, d, h, m, s (year, month, day, hour, minute, second)
  - +, -, no operator
  - For example, '-2y' (2 years ago), '-6m' (6 months ago), '8h' (at 8 AM), etc.

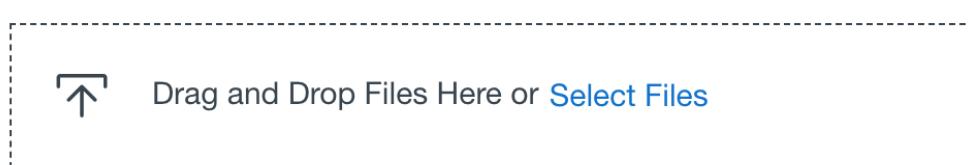
### 3.3 File Upload maxSize

```
"componentActions": [{
 "name": "upload",
 "actionType": "upload",
 "property": {
 "multiple": true,
 "accept": "image/*, .pdf, text/plain",
 "maxSize },
 ...
}]
```

### 3.3 labelTemplate of Action (Amex Only)

```
{ "name": "uploadBtn",
 ...
 "appearance": {
 "labelTemplate": "<div style=\"cursor: auto\"><div class=\"pad-2\" style=\"border-width: 1px; border-style: dashed\">Drag and Drop Files Here or <span
 class=\"placeholder\"></div><p class=\"pad-1-tb\">Files should be in PDF,
 JPG, or PNG format and cannot exceed 10MB</p></div>",
 "style": "tertiary",
 "label": "Select Files"
 }, ...
}
```

The real button is inserted inside the tag with class "placeholder".



Files should be in PDF, JPG, or PNG format and cannot exceed 10MB

## 3.3 loaded Handler

- loaded handler in generalHandlers is triggered when a component finishes loading its data or a content finishes loading its content.

```
"generalHandlers": {
 "loaded": {
 "assignments": [...],
 "updates": [...],
 ...
 }
}
```

## 3.3 More styleclassNames

```
"appearance": {
 ...
 "stylingClassNames": {
 "actions": {
 "default": ...,
 "actionName1": ...
 "actionName2": ...
 }
 }
}
```

## 3.3 More styleclassNames

- actions specific
- records
  - The wrapper of all the records
  - Different from record, which add classes to individual record
- mint and ext only

```
"appearance": {
 ...
 "stylingClassNames": {
 "records": "..."
 "actions": {
 "default": ...,
 "componentActionsTop": ...,
 "componentActionsBottom": ...,
 "componentActionsTopLeft": ...,
 "componentActionsTopRight": ...,
 "componentActionsBottomLeft": ...,
 "componentActionsBottomRight": ...,
 "recordActionsTop": ...,
 "recordActionsBottom": ...,
 "recordActionsTopLeft": ...,
 "recordActionsTopRight": ...,
 "recordActionsBottomLeft": ...,
 "recordActionsBottomRight": ...,
 "actionName1": ...,
 "actionName2": ...
 }
 }
}
```

### 3.3 Document Upload Enhancement

- Each document has its own progress bar, complete status, and the error message.
  - When a document uploading is in progress, it can be cancelled.
  - A document error message can be closed.

The screenshot shows a user interface for document upload. At the top right are two green buttons: "Foo Action" and "Upload File". Below them is a table with two columns: "File Name" and "File Size".

File Name	File Size
SKO Consumer Demo Flows UPDATED.pdf	<span style="color: red;">⚠ You cannot upload files whose size is larger than 10 MB.</span> <span style="float: right;">×</span>
Lithium Release for AmEx.pdf	<div style="width: 20%;">40%</div>   Cancel
Thrive White.png	Completed

### 3.3 Document Upload Enhancement

		FOO ACTION	UPLOAD FILE
File Name	File Size		
UI Components.docx		X	
	You can only upload images or PDF files.		
List of all portal projects.xlsx		X	
	You can only upload images or PDF files.		
Lithium Release for AmEx.pdf		29%   Cancel	
Thrive White.png		Completed	

### 3.3 Document Upload Enhancement

Mint:

```
"settings": {
 "uploadText": {
 "completed": "Completed",
 "cancelInProgress": "Cancel",
 "removeError": ""
 }
}
```

Mui: removeError has to be the name of a material UI icon.

```
"settings": {
 "uploadText": {
 "completed": "Completed",
 "cancelInProgress": "Cancel",
 "removeError": "close"
 }
}
```

### 3.3 Extension for render, enable, editable

```
"appearance": {
 "render": {
 "rule": "$ext.inYear",
 "arguments": ["$state(FeeCards)[0].attr(clcommon_Date_c) ;", 2019]
 }
}
```

```
<script type="text/javascript">
((portalext, $) => {
 portalext.inYear = (curr_date, year) => {
 if (curr_date === undefined) { return false; }
 return moment(curr_date).year() === year;
 };
})((window.portalext = window.portalext || {}), jQuery);
</script>
```

### 3.3 Extension for Remote Action Processing

- Pre-processing is to process the data before it is sent to the remote API.
- Post-processing is to process the data immediately after it is received as the remote API response. Then data can be consumed by the portal framework.

```
"componentActions" / "recordActions" {
 ...
 "remoteActionClassname": "...",
 "remoteActionArguments": { ... },
 "remoteActionProcess": {
 "pre": "$ext.preProcessMethod",
 "post": "$ext.postProcessMethod"
 },
 ...
}
```

### 3.3 Reference Field Enhancement (Mui only)

- Reference field can be displayed as select buttons.

```
"fields": [{
 "id": "...",
 "fieldName": "...",
 "relationType": "reference",
 ...
 "fieldUi": {
 "fieldType": "selectbuttons",
 ...
 }
}
```

### 3.3 selectbuttons / radiobuttons support grid (Mui only)

C-Corporation	S-Corporation
Limited Liability Company	Sole Proprietorship
Partnership	Non-Profit
Independent Contractor	Self-Employed

Is this owner a guarantor on this loan?\*

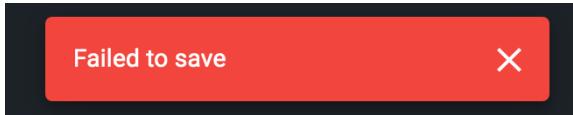
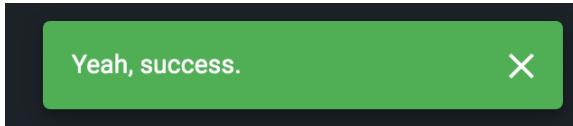
Yes       No

```
"appearance": {
 "grid": {
 "field": {
 "fieldName": {
 "selectButton": { "sm": 6 }
 }
 }
 }
}

"appearance": {
 "grid": {
 "field": {
 "fieldName": {
 "radioButton": { "sm": 6 }
 }
 }
 }
}
```

### 3.3 Remote Message Enhancement (Mui only)

- Remote success / error message using snackbar can be dismissed.



## 3.3 Date Field Enhancements

Safari-compatible "date field"

- Safari treats type=date input field as a plain text input field.
- SFDC date value is in the YYYY-MM-DD format
- When the portal frame detects the Safari browser and renders a date field, it uses the field-level configuration to convert the date value from the SFDC format to a more user-friendly format, e.g.,

```
"fieldUi": { "settingsOptions": { "fieldFormat": "MM/DD/YYYY" } }
```

- Save action converts the date string typed by a user to the SFDC format automatically before it sends to the backend because it uses the actor structure and can thus access the fieldFormat configuration.
- Submit action, on the other hand, needs each date field in the remoteActionArguments to be explicitly configured as follows:

```
"remoteActionArguments": { "account": { "expiry":
"$this.attr(SLAExpirationDate__c, sfodate, 'MM/DD/YYYY');", ...
```

## 3.3 Date Field Enhancements (cont.)

### Read-only format

- By using `fieldUi: settingsOptions: fieldFormat`, the read-only date value can be presented in a user-friendly format as well
- Can be used to get rid of the time-zone issue.

## 3.3 Date Field Enhancements (cont.)

### Built-in date field validation (without ext JS)

- **\$date**

```
"validation": [{
 "rule": "$date",
 "date": {
 "isFutureDate": boolean,
 "isPastDate": boolean,
 "includeToday": boolean,
 "startYear": number | string,
 "endYear": number | string,
 },
 "message": "Invalid date"
}] ,
```

- `startYear` can be an absolute year as a number or a relative year value as a string ("+2", or "-5")
- `endYear` can be an absolute year as a number or a relative year value as a string ("+2", or "-5")
- If not specified, `startYear` will be 1700 and `endYear` will be 4000 (according to SFDC date range)

## 3.3 Date Field Enhancements (cont.)

Built-in age validation (without ext JS)

- \$age

```
"validation": [{
 "rule": "$age",
 "age": {
 "minAge": number,
 "maxAge": number,
 },
 "message": "Invalid age"
}] ,
```

### 3.3 "Local" variant for Mui actions

```
[{
 "name": "next",
 "appearance": { "variant": "text", "label": "Next", "style": "primary" },
 ...
, {
 "name": "reload",
 "appearance": { "variant": "outlined", "label": "Reload", "style": "primary" },
 ...
, {
 "name": "redirect",
 "appearance": { "label": "Redirect", "style": "primary" },
 ...
}]
```

The default variant is "contained".

NEXT

RELOAD

REDIRECT

### 3.3. Reload browser

- "redirect": "\$self;"

### 3.3 Hide Step Titles

- "appearance": {  
    "hideTitles": true  
}

## 3.3 Range Enhancement in Mui

- Support tickmarks (check slide 195-197 for configuration details) and value bubble (Mui only)

```
"fieldUi": {
 "fieldType": "range",
 "label": "Lorem ipsum",
 "tooltip": "Lorem ipsum dolor sit amet, te eos error recteque",
 "rangeOptions": {
 "dataType": "percent",
 ...
 "editableRangeValue": true,
 "valueLabelDisplay": true
 },
 "settingsOptions": {"minimumFractionDigits": 0},
 "format": {"options": {"numeral": true, "numeralDecimalScale": 0}}
}
```



Lorem ipsum dolor sit amet, te eos error recteque dignissim, sed oporteat efficiantur definitionem te.



## 3.3 Google Address Complete Enhancement

- addressTypes
  - [https://developers.google.com/places/supported\\_types#table3](https://developers.google.com/places/supported_types#table3)
  - By default, if nothing is specified, the addressTypes are ["address"] because address instructs the Place Autocomplete service to **return only geocoding results with a precise address.**
  - All the values listed in the above link can be used.

```
{
 "fieldUi": {
 "fieldType": "addressAutocomplete",
 "addressComponentMapping": [...],
 "addressComponentRestrictions": {...},
 "addressTypes": ["geocode"]
 }
}
```

## 3.3 Amex Tooltip Enhancement

- Support tooltip placement
  - top, bottom, left, right
  - right is the default placement
- Syntax:
  - `title: text{{{tooltip}}}` **position**
  - `tooltip: {{{tooltip}}}` **position**

## 3.4 Portal Enhancements

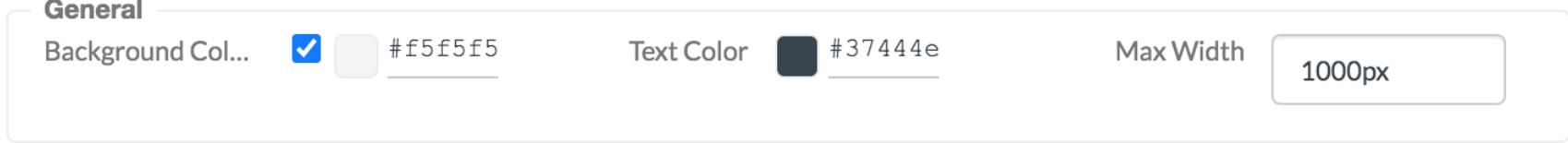
- Responsive portal
- Portal help and syntax autocompletion

# Assumptions

- Single navigation structure for both desktop and phone
- Changes in look and feel only
  - Layout
  - View rendering
  - Stepper (Mui)
  - Field layout (Mui)
  - Certain types of field (Mui)
    - range
  - Dialog (Mui)
  - Text
  - Buttons

# Layout

- maxWidth



- Phone (CSS enforcement)
  - No sidebar
  - Single column only
  - Global scrolling

# Conditional array and \$client;

```
[
 {
 "condition": "$client; == \"phone\\\"",
 "value": "..."
 },
 {
 "condition": "$client; != \"phone\\\"",
 "value": "..."
 },
 ...
]
```

# View Rendering

```
"appearance": {
 "renderAs": [
 {
 "condition": "$client; == \"phone\"",
 "value": "card-view"
 },
 {
 "condition": "$client; != \"phone\"",
 "value": "list-view"
 }
]
}
```

## Existing Accounts – Total Revenue: \$6,748,030,753.00

Account Name ↑	Industry	Account Phone	↓ Annual Revenue
United Oil & Gas Corp.	Energy	(212) 842-5500	\$ 5,470,000,000.00
Dickenson plc	Consulting	(785) 241-6200	\$ 500,000,000.00
Grand Hotels & Resorts Ltd	Insurance	(312) 596-1004	\$ 500,000,000.00
Edge Communications	Electronics	(512) 757-6000	\$ 139,000,000.00
Express Logistics and Transport 2	Insurance	(503) 421-7800	\$ 95,000,000.00
GenePoint 2	Biotechnology	(650) 867-3450	\$ 30,000,000.00
Pyramid Construction Inc.	Construction	(014) 427-4427	\$ 9,500,000.00

## Existing Accounts

Sort By Annual Revenue (High to Low) ▾

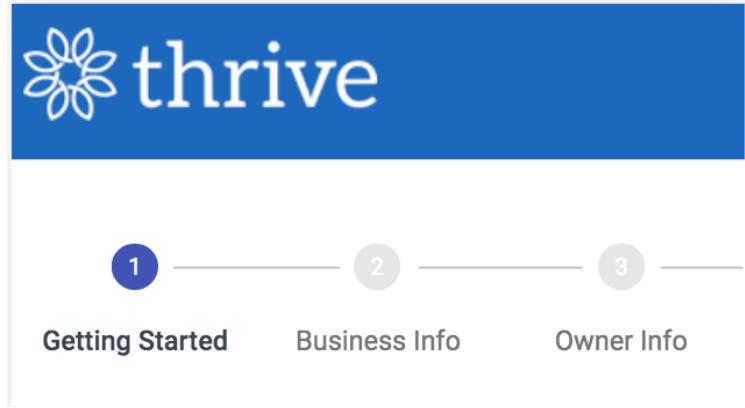
Account Name	United Oil & Gas Corp.
Industry	Energy
Account Phone	(212) 842-5500
Annual Revenue	\$5,470,000,000.00

Account Name	Dickenson plc
Industry	Consulting
Account Phone	(785) 241-6200
Annual Revenue	\$500,000,000.00

Account Name	Grand Hotels & Resorts Ltd
Industry	Insurance
Account Phone	(312) 596-1004
Annual Revenue	\$500,000,000.00

# Stepper

```
"appearance": {
 "titles": [],
 "numberOfVisibleTitles": [
 {
 "condition": "$client; == \"phone\"",
 "value": 3
 }
]
}
```



# Field Layout

```
{
 "sm": 6,
 "xs}
```

# Range Field

Lorem ipsum  
70

%  
0% ————— 100%

Lorem ipsum dolor sit amet, te eos error recteque dignissim, sed oporteat efficiantur definitionem te.

Market Value  
\$ 365,000

\$120,000 ————— \$1,000,000

Loan Amount  
\$ 75,000

\$75,000 ————— \$255,499  
Market Value 50% 60% 70%

Lorem ipsum

70

%

70

0%

100%

Lorem ipsum dolor sit amet, te eos error recteque dignissim, sed oporteat efficiantur definitionem te.

Market Value  
\$ 120,000

\$120,000 ————— \$1,000,000

Loan Amount  
\$ 75,000

\$75,000 ————— \$84,000  
Market Value 70%

# Dialog

- In phone, overlay modal dialog takes full screen.

Collateral Information X

Cloud icon

Collateral Type \* Equipment

Description \*

Estimated Value \* \$

CANCEL SAVE

# Text Related

```
"appearance": {
 "title": [
 {
 "condition": "$client; == \"phone\"",
 "value": "Existing Accounts"
 },
 {
 "condition": "$client; != \"phone\"",
 "value": "Existing Accounts — Total Revenue: $this.sum(AnnualRevenue,
currency);"
 }
]
}
```

# Buttons

```
"appearance": [{
 "value": {
 "label": "Next",
 "style": "primary",
 "position": "top-right"
 },
 "condition": "$client; != \"phone\\""]
, {
 "value": {
 "style": "iconic",
 "label": "share",
 "position": "top-right"
 },
 "condition": "$client; == \"phone\\""
}]

"context": [
 {
 "condition": "$client; ==
 \"phone\\"",
 "value": "phoneGroup"
 }
]
```

## Custom Marks

FOO FOO

BAR BAR

BAZ BAZ

NEXT

## Custom Marks



## Custom Marks

Lorem ipsum

70

Foo Foo

Bar Bar

Baz Baz

6

## 3.4 Assignment with recordId

- Assign value to a specific record (not just the first record) in a list/queue/card view

First Name	Last Name	Business Phone	Account ID	
Jake	Llorrac		sForce	
Rose	Gonzalez	(512) 757-6000	Edge Communications	
Sean	Forbes	(512) 757-6000	Edge Communications	
Jack	Rogers	(336) 222-7000	Times of India	



```
"assignments": [{ "field": "$global.attr(contactId) ;", "value": "$this.attr(id) ;" }]
```

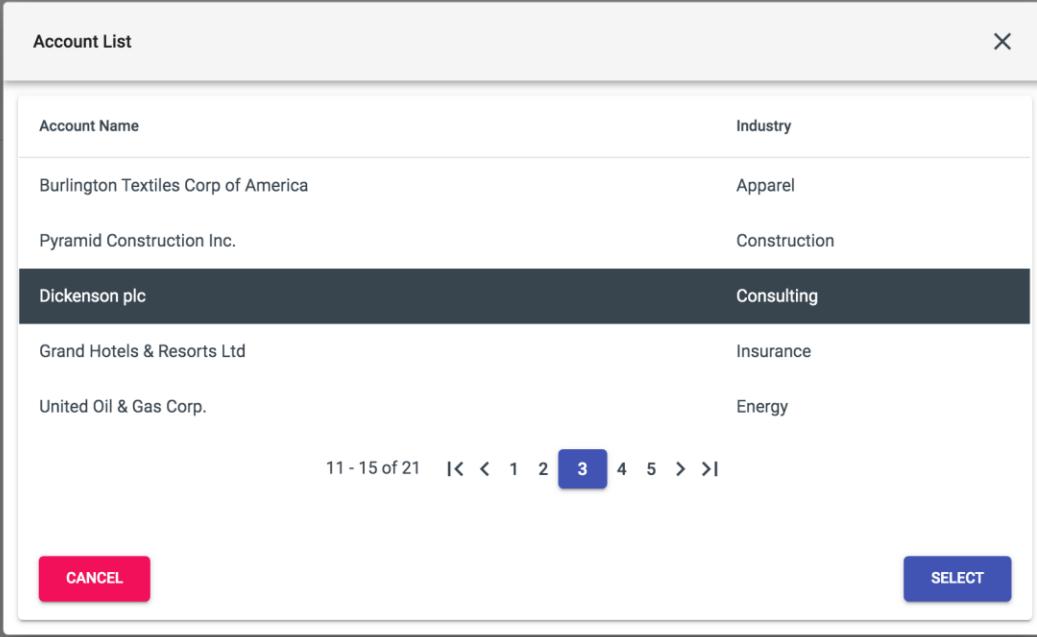
# 3.4 Assignment with recordId

Account List

Account Name	Industry
Burlington Textiles Corp of America	Apparel
Pyramid Construction Inc.	Construction
Dickenson plc	Consulting
Grand Hotels & Resorts Ltd	Insurance
United Oil & Gas Corp.	Energy

11 - 15 of 21 | < < 1 2 3 4 5 > >|

**CANCEL** **SELECT**



```
"assignments": [{
 "field": "$state(ContactList).attr(AccountId);",
 "value": "$interaction.attr(highlighted).attr(id);",
 "recordId": "$global.attr(contactId);"}] ,
```

## 3.4 Disable browser autocomplete and cut-copy-paste

- fieldUi: {  
    "disableCopyPaste": true,  
    "disableAutocomplete": true  
}

## 3.4 Add conditional entries to regex

```
{
 "fieldName": "phoneType",
 "fieldUi": {
 "fieldType": "picklist",
 "label": "Phone Type",
 "placeholderText": "Select One...",
 "selectOptions": [{
 "optionText": "Mobile",
 "optionValue": "Mobile"
 }, {
 "optionText": "Landline",
 "optionValue": "Landline"
 }, {
 "optionText": "Other",
 "optionValue": "Other"
 }]
 }
}
```

```
{
 "fieldName": "phoneNumber",
 "fieldUi": {
 "fieldType": "string",
 "label": "Phone Number",
 "required": true,
 "regex": [{
 "condition":
 "$this.attr(phoneType) == \"Mobile\"",
 "value": "^\\d{8}$"
 }, {
 "condition":
 "$this.attr(phoneType) == \"Landline\"",
 "value": "^\\d{7}$"
 }]
 }
}
```

## 3.4 Tree View

- Tree view can be rendered from a list of records, as long as each record has parentId.
- parentId can be undefined. In this case, the record is a root.

- ▼ Application Documents
  - ▼ Business Entity
    - Business License
    - **Business Tax Return**
    - Corporate Documents

## 3.4 Tree View

```
{
 "tableName": "clcommon_document_category_c",
 "whereClause": {
 "freeForm": "genesis_Application_c = ':applicationId' AND
(clcommon_Parent_Document_Category_c = null OR
(clcommon_Parent_Document_Category_c != null AND clcommon_Show_In_Portal_c
= 'UPLOAD_VIA_PORTAL'))"
 },
 "sections": [{
 "fields": [{
 "fieldName": "clcommon_Category_Name_c",
 "fieldUi": { ... }
 }, {
 "fieldName": "clcommon_Parent_Document_Category_r.Id",
 "fieldUi": { }
 }]
 }]
}
```

## 3.4 Tree View

```
"clickRecord": {
 "assignments": [{
 "field": "$global.attr(categoryId);",
 "value": "$this.attr(id);"
 }]
}

{
 "treeExpand": true,
 "treeViewMapping": {
 "entry": "clcommon_Category_Name_c",
 "parentId": "clcommon_Parent_Document_Category_r.Id"
 },
 "renderAs": "tree-view"
}
```

- `treeViewMapping` converts a flat list of records into a hierarchical tree structure
- Each tree node has `entry` (as label of a tree node) and `children`, as well as other attributes that can be used by handlers
- Only the leaf nodes can be clicked and highlighted.
- By default, all roots are collapsed.
- If `treeExpand` is true, all the tree nodes are expanded by default.

## 3.4 Tree View

- Tree view can work with staticData
  - entry and children are must-have.

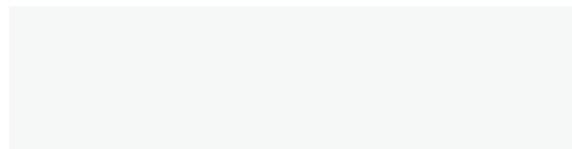
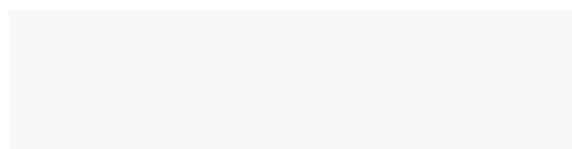
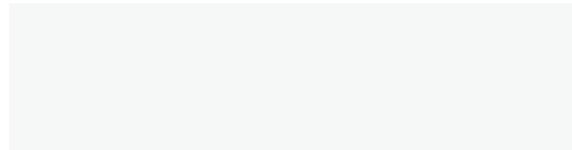
```
"staticData": [{
 "entry": "Application Documents",
 "categoryId": "a1K4S000000p2GdUAI",
 "children": [{
 "entry": "Business Entity",
 "categoryId": "a1K4S000000p2GeUAI",
 "children": [{
 "entry": "Business License",
 "categoryId": "a1K4S000000p2GfUAI"
 }, {
 "entry": "Business Tax Return",
 "categoryId": "a1K4S000000p2GhUAI"
 }, {
 "entry": "Corporate Documents",
 "categoryId": "a1K4S000000p2GgUAI"
 }]
 }]
}]
```

# 3.4 Skeleton

- Portal actor can display a placeholder preview when it is loading

## Required Documents

The following documents are needed to process your application.



SUBMIT

## Required Documents

The following documents are needed to process your application.

### Business License OPEN

Federal license or permit if your business activities are regulated by a federal agency. Business License from the state, county, or city depending on your business activities and business location



### Business Tax Return OPEN

Financial Statements and/or annual income tax return. The form you use depends on how your business is organized. This could be form 1020S,1040,1065



### Corporate Documents OPEN

Certificate of Incorporation, Memorandum of Association, Articles of Association, Bylaws and any other corporate document, including any shareholders, voting and/or any other agreement or document relating to the incorporation, ownership or management of an entity



SUBMIT



Category Attachment Name	Attachment Id	Record ID
--------------------------	---------------	-----------

[REDACTED]	[REDACTED]	[REDACTED]
------------	------------	------------



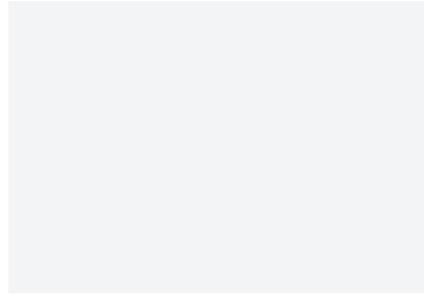
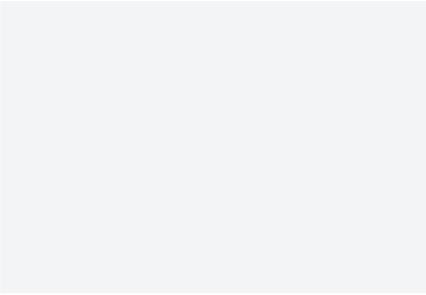
Category Attachment Name	Attachment Id	Record ID
--------------------------	---------------	-----------

secured_smb_flow_responsive.txt	00P4S000000pkvFUAQ	a1K4S000000p2GfUAI
secured_smb_portal_responsive - 6-22-2020 4-58 PM.txt	00P4S000000pkvAUAQ	a1K4S000000p2GfUAI
bounce_back_loan - 5-1-2020 10-19 PM.txt	00P4S000000pku2UAA	a1K4S000000p2GfUAI

## Ownership

ADD OWNER

Add all individuals with a 20% or greater ownership stake



BACK

NEXT

ADD OWNER

## Ownership

Add all individuals with a 20% or greater ownership stake

John Smith OWNER

1 Stoneridge Mall Road

Pleasanton, CA 94588

(321) 312-3123

john@abc.com

\*\*\*-\*\*-3213

Ownership Percentage: 10%

BACK

NEXT

How much money are you looking to borrow?



Loans above \$250,000 require collateral. Please be prepared to provide collateral information and documentation, such as proof of ownership, which are required to process your credit request.

BACK

NEXT

NEXT

How much money are you looking to borrow?

\$ 10,000

Loan amount cannot be more than \$900,000.

Loans above \$250,000 require collateral. Please be prepared to provide collateral information and documentation, such as proof of ownership, which are required to process your credit request.

BACK

NEXT

Tell us about yourself

Tell us about yourself

First Name \*

Last Name \*

Street Address \*

City \*

State \*

Select a state...

Postal Code \*

Phone Number \*

Email Address \*

## 3.4 Skeleton Global Settings

```
"skeleton": {
 "listView": {
 "withAction": {
 "height": "72px",
 "padding": "24px 12px"
 },
 "noAction": {
 "height": "48px",
 "padding": "12px"
 },
 "queueView": {
 "height": "150px",
 "padding": "16px"
 },
 "cardView": {
 "height": "300px",
 "padding": "16px"
 },
 "detailView": {
 "large": {
 "height": "108.75px",
 "padding": "28px 12px 20px 12px"
 },
 "medium": {
 "height": "104px",
 "padding": "28px 12px 20px 12px"
 },
 "small": {
 "height": "101.625px",
 "padding": "28px 12px 20px 12px"
 }
 }
 }
}
```

Specify the height and padding of the default shaded rectangle. width is either 100% or controlled by grid.

## 3.4 Skeleton Items

- Each actor needs to specify how many skeleton rectangles to be displayed.

```
"appearance": {
 "skeleton": {
 "items": 3
 }
}
```

## 3.4 Skeleton Overridden

What is the purpose of this loan?

BACK

NEXT

What is the purpose of this loan?

Working Capital	Purchase of Real Estate
Purchase Equipment	Purchase Inventory
Business Expansion	Other

BACK

NEXT

```
"appearance": {
 "skeleton": {
 "items": 6,
 "style": {
 "padding": "12px",
 "height": "74px"
 },
 "grid": {
 "sm": 6
 }
 }
}
```

## 3.4 Skeleton with renderAs

```
"appearance": {
 "renderAs": [{
 "condition": "$client; == \"phone\"",
 "value": "card-view"
 }, {
 "condition": "$client; != \"phone\"",
 "value": "list-view"
]},
 "skeleton": {
 "card-view": {
 "items": 10,
 "style": { "height": "160px" }
 },
 "list-view": {
 "items": 10
 }
 }
}
```

## 3.4 Skeleton with Conditional Entries

```
"appearance": {
 "skeleton": [{
 "value": {
 "items": 3,
 "style": { "height": "120px" }
 },
 "condition": "$client; != \"phone\\""
 }, {
 "value": {
 "items": 3,
 "style": { "height": "178px" }
 },
 "condition": "$client; == \"phone\\""
 }]
}
```

# 3.4 Vertical Stepper with Sub-steps (Mui only)

The screenshot shows a configuration interface for a 'VerticalStepper' component. It consists of three stacked panels, each containing a title and subtitle input field, a 'Name' field, a checked checkbox, a 'Render' section with a checked checkbox, and a 'Step is visible' status indicator.

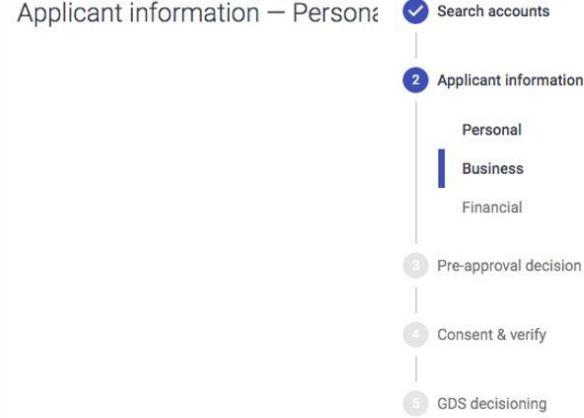
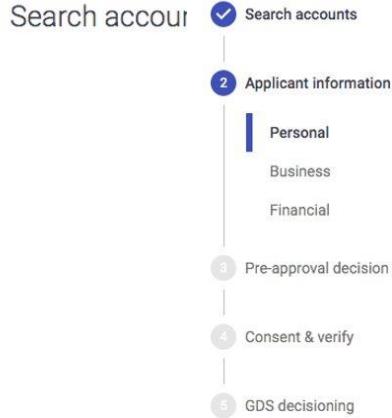
- Panel 1:** Title: 'Search accounts', Subtitle: 'Enter sub-title'. Name: 'SearchAccounts'. Step is visible.
- Panel 2:** Title: 'Applicant information', Subtitle: 'Personal'. Name: 'ApplicantPersonal'. Step is visible.
- Panel 3:** Title: 'Applicant information', Subtitle: 'Business'. Name: 'ApplicantBusiness'. Step is visible.

At the bottom, there are 'Apply' and 'Cancel' buttons.

```
"appearance": {
 "titles": [{
 "title": "Search accounts"
 }, {
 "title": "Applicant information", "subTitle": "Personal"
 }, {
 "title": "Applicant information", "subTitle": "Business"
 }, {
 "title": "Applicant information", "subTitle": "Financial"
 }, {
 "title": "Pre-approval decision"
 }, {
 "title": "Consent & verify", "subTitle": "Consent"
 }, {
 "title": "Consent & verify", "subTitle": "Verify"
 }, {
 "title": "GDS decisioning"
 }],
 "titleOrientation": "vertical", "titleWidth": "250px"
}
```

Sub-steps are still normal steps. Only change their look and feel.

- 1 Search accounts
- 2 Applicant information
- 3 Pre-approval decision
- 4 Consent & verify
- 5 GDS decisioning



- ✓ Search accounts
- ✓ Applicant information
- 3 Pre-approval decision
- 4 Consent & verify
- 5 GDS decisioning

### Pre-approval decision

- ✓ Search accounts
- ✓ Applicant information
- ✓ Pre-approval decision
- 4 Consent & verify
- 5 GDS decisioning

### Consent

# 3.4 Table with Summary (Mui only)

- List view can have the summary rows at the bottom (list view only)

Account Name ↑	Industry	Account Phone	Annual Revenue
Dickenson plc	Consulting	(785) 241-6200	\$ 500,000,000.00
Edge Communications	Electronics	(512) 757-6000	\$ 139,000,000.00
Express Logistics and Transport 2	Insurance	(503) 421-7800	\$ 95,000,000.00
GenePoint 2	Biotechnology	(650) 867-3450	\$ 30,000,000.00
Grand Hotels & Resorts Ltd	Insurance	(312) 596-1004	\$ 500,000,000.00
United Oil & Gas Corp.	Energy	(212) 842-5500	\$ 5,470,000,000.00
Total Accounts			6
Total Annual Revenue			\$6,734,000,000.00

- Each row is an array of values.
- null value indicates empty cell.
- If a value is an object, it should have a value key and an align key.

```
"appearance": {
 "summary": {
 "rows": [
 {"row": [
 "Total Accounts", null, null,
 {
 "value": "$this.count()",
 "align": "right"
 }
]},
 {"row": [
 "Total Annual Revenue", null, null,
 {
 "value":
 "$this.sum(AnnualRevenue, currency)",
 "align": "right"
 }
]}
]}
}
```

Account Name ↑	Industry	Account Phone	Annual Revenue
Dickenson plc	Consulting	(785) 241-6200	\$ 500,000,000.00
Edge Communications	Electronics	(512) 757-6000	\$ 139,000,000.00
Express Logistics and Transport 2	Insurance	(503) 421-7800	\$ 95,000,000.00
GenePoint 2	Biotechnology	(650) 867-3450	\$ 30,000,000.00
Grand Hotels & Resorts Ltd	Insurance	(312) 596-1004	\$ 500,000,000.00
United Oil & Gas Corp.	Energy	(212) 842-5500	\$ 5,470,000,000.00
Total Accounts		6	
Total Annual Revenue			\$6,734,000,000.00

Account Name ↑	Industry	Account Phone	Annual Revenue
Dickenson plc	Consulting	(785) 241-6200	\$ 500,000,000.00
Edge Communications	Electronics	(512) 757-6000	\$ 139,000,000.00
Express Logistics and Transport 2	Insurance	(503) 421-7800	\$ 95,000,000.00
GenePoint 2	Biotechnology	(650) 867-3450	\$ 30,000,000.00
Grand Hotels & Resorts Ltd	Insurance	(312) 596-1004	\$ 500,000,000.00
United Oil & Gas Corp.	Energy	(212) 842-5500	\$ 5,470,000,000.00
Total Accounts		6	
Total Annual Revenue			\$6,734,000,000.00

Account Name ↑	Industry	Account Phone	Annual Revenue
Dickenson plc	Consulting	(785) 241-6200	\$ 500,000,000.00
Edge Communications	Electronics	(512) 757-6000	\$ 139,000,000.00
Express Logistics and Transport 2	Insurance	(503) 421-7800	\$ 95,000,000.00
GenePoint 2	Biotechnology	(650) 867-3450	\$ 30,000,000.00
Grand Hotels & Resorts Ltd	Insurance	(312) 596-1004	\$ 500,000,000.00
United Oil & Gas Corp.	Energy	(212) 842-5500	\$ 5,470,000,000.00
Total Accounts		6	
Total Annual Revenue			\$6,734,000,000.00

```
"summary: {
 "prefixSpan": 2,
 "rows": [
 { "row": [
 "Total Accounts", {
 "value": "$this.count();",
 "align": "right"
 }
] ...] }
```

```
"summary: {
 "suffixSpan": 2,
 "rows": ... }
```

```
"summary: {
 "prefixSpan": 1,
 "suffixSpan": 1,
 "rows": ... }
```

<input type="checkbox"/>	Account Name ↑	Industry	Account Phone	Annual Revenue
<input type="checkbox"/>	Dickenson plc	Consulting	(785) 241-6200	\$ 500,000,000.00
<input type="checkbox"/>	Edge Communications	Electronics	(512) 757-6000	\$ 139,000,000.00
<input type="checkbox"/>	Express Logistics and Transport 2	Insurance	(503) 421-7800	\$ 95,000,000.00
<input type="checkbox"/>	GenePoint 2	Biotechnology	(650) 867-3450	\$ 30,000,000.00
<input type="checkbox"/>	Grand Hotels & Resorts Ltd	Insurance	(312) 596-1004	\$ 500,000,000.00
<input type="checkbox"/>	United Oil & Gas Corp.	Energy	(212) 842-5500	\$ 5,470,000,000.00
Total Accounts				6
Total Annual Revenue				\$6,734,000,000.00

Use "prefixSpan": 1 for the multi-selection check box column.

```
"summary": {
 "prefixSpan": 1,
 "rows": [
 {"row": [
 "Total Accounts",
 null,
 null,
 {
 "align": "right",
 "value": "$this.count();"
 }
]},
 {"row": [
 "Total Annual Revenue",
 null,
 null,
 {
 "align": "right",
 "value": "
 $" + this.sum(AnnualRevenue, currency);
 }
]}
}
```

```
"appearance": {
 "renderFields": {
 "Phone": "$global.attr(showPhone) == true"
 },
 "summary": [{
 "value": {
 "prefixSpan": 1,
 "rows": [
 {"row": ["Total Accounts", null, null, {"align": "right", "value":
"$this.count();"}]},
 {"row": ["Total Annual Revenue", null, null, {"align": "right", "value":
"$this.sum(AnnualRevenue, currency);"}]
]
 },
 "condition": "$global.attr(showPhone) == true"
 }, {
 "value": {
 "prefixSpan": 1,
 "rows": [
 {"row": ["Total Accounts", null, {"align": "right", "value": "$this.count();"}]},
 {"row": ["Total Annual Revenue", null, {"align": "right", "value":
"$this.sum(AnnualRevenue, currency);"}]
]
 },
 "condition": "$global.attr(showPhone) != true"
 }]
}
```

summary supports conditional entry.

<input type="checkbox"/> Account Name ↑	Industry	Account Phone	Annual Revenue
<input type="checkbox"/> Dickenson plc	Consulting	(785) 241-6200	\$ 500,000,000.00
<input type="checkbox"/> Edge Communications	Electronics	(512) 757-6000	\$ 139,000,000.00
<input type="checkbox"/> Express Logistics and Transport 2	Insurance	(503) 421-7800	\$ 95,000,000.00
<input type="checkbox"/> GenePoint 2	Biotechnology	(650) 867-3450	\$ 30,000,000.00
<input type="checkbox"/> Grand Hotels & Resorts Ltd	Insurance	(312) 596-1004	\$ 500,000,000.00
<input type="checkbox"/> United Oil & Gas Corp.	Energy	(212) 842-5500	\$ 5,470,000,000.00
Total Accounts			6
Total Annual Revenue			\$6,734,000,000.00

<input type="checkbox"/> Account Name ↑	Industry	Annual Revenue
<input type="checkbox"/> Dickenson plc	Consulting	\$ 500,000,000.00
<input type="checkbox"/> Edge Communications	Electronics	\$ 139,000,000.00
<input type="checkbox"/> Express Logistics and Transport 2	Insurance	\$ 95,000,000.00
<input type="checkbox"/> GenePoint 2	Biotechnology	\$ 30,000,000.00
<input type="checkbox"/> Grand Hotels & Resorts Ltd	Insurance	\$ 500,000,000.00
<input type="checkbox"/> United Oil & Gas Corp.	Energy	\$ 5,470,000,000.00
Total Accounts		6
Total Annual Revenue		\$6,734,000,000.00

## 3.4 fieldChangeCommit handler

- range field only
- The handler will be triggered when a user drags the thumb of a slider and release the mouse. The handler can trigger actions.

```
"generalHandlers": {
 "fieldChangeCommit": {
 "loanAmount": {
 "actions": [
 {
 "name": "submitLoanAmount"
 }
]
 }
 }
}

 "fields": {
 "fieldName": "loanAmount",
 "fieldUi": { "fieldType": "range", ...
 "componentActions": [{
 "name": "submitLoanAmount",
 "actionType": "submit",
 }
]
}
```

## 3.4 Action Extraction Enhancements

- Actor property does not include remote-action-related any more.  
Instead it refers to a query action using the `queryName`.
- New `actionType`
  - `query`
  - No appearance because it won't be rendered
  - No handlers
- Actor `query` refers to `componentAction`

```
"property": {
 "structureName": "...",
 "queryName": "query1"
}

"componentActions": [{
 "name": "query1",
 "actionType": "query",
 "remoteActionArguments": {...},
 "remoteActionClassname":
 "PortalGetAccountAPI"
}]
```

## 3.4 Action Extraction Enhancements

- Reference query refers to recordAction

```
"property": {
 "structureName": "...",
 "referenceQueryName": {
 "Broker_Name__r": "query1"
 }
}

"recordActions": [{
 "name": "query1",
 "actionType": "query",
 "remoteActionArguments": {...},
 "remoteActionClassname":
 "PortalGetReferenceAPI"
}]
```

Match the  
relationName value

In order to call remote API,  
relationJSON should NOT  
have tableName and  
whereClause!

```
{
 "structureName": "...",
 "sections": [{
 "fields": [{
 "relationType": "reference",
 "fieldName": "Broker_Name__c",
 "relationName": "Broker_Name__r",
 "relationJson": {
 "sections": [...]
 },
 "fieldUi": {
 "fieldType": "reference",
 "label": "Broker Name"
 }
 }]
 }]
}
```

## 3.4 Document Upload Enhancements

- Attachment is deprecated by SFDC. Therefore we recommend to use ContentDocument to store uploaded documents.

```
componentActions: [{
 "property": {
 "uploadType": "document"
 }
}]
```

## 3.4 Document Upload Enhancements

- If "uploadType": "document", the following three property keys can be added
  - contentLocation
    - "**S    - "E": E-Document is located outside the Salesforce.
    - "L": L-Document is located on a social network and accessed via Social Customer Service.**
  - visibility
    - "**AllUsers    - "InternalUsers": The file is available only to internal users who have permission to see the file.
    - "SharedUsers": The file is available to all users who can see the feed to which the file is posted.**
  - shareType
    - "**V**" (Viewer permission): The user can view, download and share the file. (default value)
    - "C" (Collaborator permission): The user can view, download, share the file, edit the details, upload new version and change sharing settings.
    - "**I**" (Inferred permission): The user's permission is determined by the related record.

## 3.5 Document Upload Error Message

- With document uploaded as attachment, the error message from the RESTful API can be displayed in the UI.

```
"componentActions": [{
 "name": "uploadDocsRacv",
 "actionType": "upload",
 "property "messages": {
 "uploadFailureText "submitFailureText": "$serverMessage;"
 }
}]
```

File Name	Status
No document found. Click upload or Drag and drop a document	
<input type="file"/> Drag and drop a document here	
<b>Upload</b>	

tsconfig.json  
File type is invalid. Please upload a file with an accepted file extension

## 3.5 Configurable Loading Page (Mui only)

- In global settings, users can specify the loadingPageUrl to customize an <iframe> as the global loading animation for save, submit, and query
  - "settings": { **loadingPageUrl**:  
  "/apex/clcommon\_\_PortalLoadingPage", ... }
- A common use case is to make the global loading animation and the animation in the splash page consistent by using the same VF page.

## 3.5 Splash Page Configuration

- Custom Visualforce Components can be used as Portal Splash pages
- Example:

CustomPortalSplashPage is a Custom Visualforce Component created locally within the Salesforce org and can be used as a Splash page as shown below.

Be sure to set showDefaultLoadingPage as FALSE

```
<c:IncludePortal2 navName="playground_loading" theme="mui" showDefaultLoadingPage="false">
 <c:CustomPortalSplashPage></c:CustomPortalSplashPage>
</c:IncludePortal2>
```

## 3.5 Action Layout

- Mint and Ext only
- In the "appearance" section
  - componentTop: true - title and subtitle, if defined, are above the component content in the HTML flow.
  - componentTop: [actionName] - componentTop: true + the listed component actions will be displayed side-by-side above the component content in the HTML flow.
  - componentBottom: [actionName] - the listed component actions will be displayed side-by-side below the component content in the HTML flow.
  - recordTop: [actionName] - the listed record actions will be displayed side-by-side above the component content in the HTML flow.
  - recordBottom: [actionName] - the listed record actions will be displayed side-by-side below the component content in the HTML flow.

## 3.5 Actor Validation Enhancement

- For `save` and `submit` actions, they validate their own actor (the actor that contains those actions) by default before the remote API is called. If there are any field validation errors in the actor, the first invalid field will be scrolled to be visible.
- For any action object, the `validation` key can be added, which takes an array of the actor names so that when the action is triggered, the data in those actors will be validated. If there are any field validation errors in the listed actors, the first invalid field **in the first listed actor** will be scrolled to be visible.

## 3.5 Dynamic Container

- Mint only
- Dynamic container can receive a list of records from another actor.
  - The actor type can be data-model so that it is not visible but can query data from the backend and pass the data to a dynamic container
  - "generalHandlers": {  
    "loaded": {  
        "assignments": [ {  
            "field": "\$state(AccountContainer);",  
            "value": "\$this;"  
        } ]  
    }  
}
- Reloading the "source" actor will re-render the dynamic container.

# 3.5 Dynamic Container (cont.)

- Dynamic container must have "dynamic": true in "appearance"
- Dynamic container only has a single child division with its title. But each record displays a section.
  - "appearance": {  
    "titles": [ {  
        "title": "\$this.attr('Name') ;  
 &mdash; \$this.attr('Industry');"  
    }],  
    "dynamic": true  
}
  - `$this.attr()` uses the data from the specific record.

The screenshot shows a configuration interface for an 'AccountContainer'. The top right corner displays the title 'AccountContainer'. The interface is divided into several sections:

- Name \***: AccountContainer
- Type \***: Accordion
- Render**:  
 Conditionally render tabs, steps or accordion
- Visible Indexes**: I,II,III,IV
- Titles and Child Actors**:  
\$this.attr('Name'); &mdash; \$this.attr('Industry')  
accountWrapper
- Name**:
- Render**:  
 Accordion is visible.
- Enable**:  
 Accordion is enabled.

At the bottom right are 'Apply' and 'Cancel' buttons.

## 3.5 Dynamic Container (cont.)

- Inside the child division, actors can be added, which will be repeated in the section of each record.
  - For example, if `ContactList` is added to the child division, it repeats in each dynamic section, as shown in the next slide. Note that the actor bar displays "ContactList" in each section. Clicking any actor bar opens the same configuration panel/dialog for `ContactList`.
  - However, the `ContactList` in each section has its unique name by appending the record data id after the common name, for example, `ContactList:001f400000wBREzAAO`. The unique actor name is not displayed in the actor bar but can be used in the configuration, for example, reloading `ContactList` in a specific section.

```
"updates": [{
 "name": "$nav.reloads.add;",
 "value": ["ContactList:$global.attr(selectedAccountId) ;"]
}
```

- Actors in the child division can refer to the record data using `$parent.attr()`.

## component structure

```
{
 "structureName": "ContactList",
 "tableName": "contact",
 "whereClause": {
 "freeForm": "account.id = ':accountId'"
 },
 "sections": [...]
}
```

The image displays three separate instances of a UI component, each showing a contact list. The component has a header with 'Edit' and a green 'Add Contact' button. Below the header is a table with columns: First Name, Last Name, and Business Phone. A red box highlights the 'Last Name' column header. The 'ContactList' placeholder is visible in the first two instances. The third instance shows a single contact entry: Eric Smith.

First Name	Last Name	Business Phone
		No Contacts
<button>Add Contact</button>		

First Name	Last Name	Business Phone
		No Contacts
<button>Add Contact</button>		

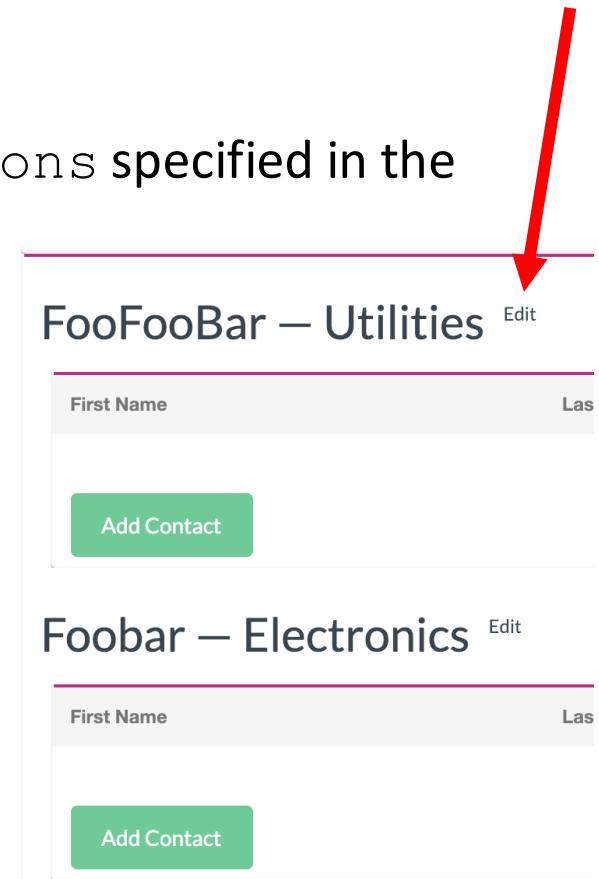
First Name	Last Name	Business Phone
Eric	Smith	
<button>Add Contact</button>		

```
"property": {
 "structureName": "ContactList",
 "whereClause": {
 "accountId": "$parent.attr(id);"
 }
},
"componentActions": [{
 "name": "addContact",
 "handler": {
 "assignments": [{
 "field":
 "$global.attr(selectedAccountId);",
 "value": "$parent.attr(id);"
 }],
 "updates": [{
 "name": "$nav.overlay.add;",
 "value": "addContactOverlay"
 }]
 }]
}]
```

## 3.5 Dynamic Container (cont.)

- Each dynamic section can have recordActions specified in the dynamic container.

```
"recordActions": [{
 "name": "editAccount",
 "handler": {
 "assignments": [{
 "field": "$global.attr(selectedAccountId);",
 "value": "$this.attr(id);"
 }],
 "updates": [{
 "name": "$nav.overlay.add;",
 "value": "editAccountOverlay"
 }]
 }]
}]
```



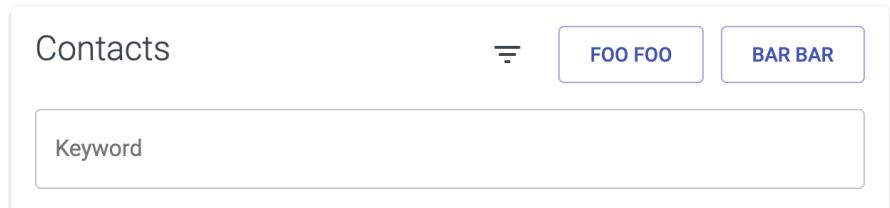
## 3.5 Dynamic Container (cont.)

- Dynamic sections can be expanded and collapsed independently.

```
{
 "titles": [{
 "title": "$this.attr(Name) — $this.attr(Industry)"
 }],
 "dynamic": true,
 "entryCollapse": {
 "collapsed": false,
 "collapsibleIcons": {
 "expanded": "",
 "collapsed": ""
 }
 }
}
```

# 3.5 Filtering

- Global setting



```
"filters": {
 "keywordLabel": "Keyword",
 "dialogTitle": "Filters",
 "icon": "filter_list",
 "label": "Filtered by",
 "clearAllText": "Clear All",
 "startDateLabel": "Start date",
 "endDateLabel": "End date",
 "minNumberLabel": "Minumum",
 "maxNumberLabel": "Maximum",
 "toLabel": "to",
 "fromLabel": "from",
 "orLabel": "or",
 "clearLabel": "Clear",
 "applyLabel": "Apply",
 "notSpecifiedLabel": "Not specified",
 "invalidText": "Invalid value"
}
```

## 3.5 Filtering (cont.)

- The global setting can be overwritten by actor appearance

```
"filters": {
 "keyword": {
 "filterLabel": "...",
 "fieldNames": [...]
 },
 "advanced": {
 "dialogTitle": "Filters",
 "icon": "filter_list",
 "label": "Filtered by",
 "clearAllText": "Clear All",
 "startDateLabel": "Start date",
 "endDateLabel": "End date",
 "minNumberLabel": "Minumum",
 "maxNumberLabel": "Maximum",
 "toLabel": "to",
 "fromLabel": "from",
 "orLabel": "or",
 "clearLabel": "Clear",
 "applyLabel": "Apply",
 "notSpecifiedLabel": "Not specified",
 "invalidText": "Invalid value",
 "filterFields": [...]
 }
}
```

## 3.5 Filtering (cont.)

- Keyword filtering is against text-base fields

```
"filters": {
 "keyword": {
 "fieldNames": [
 "Name",
 "Account.Name"
]
 }
}
```

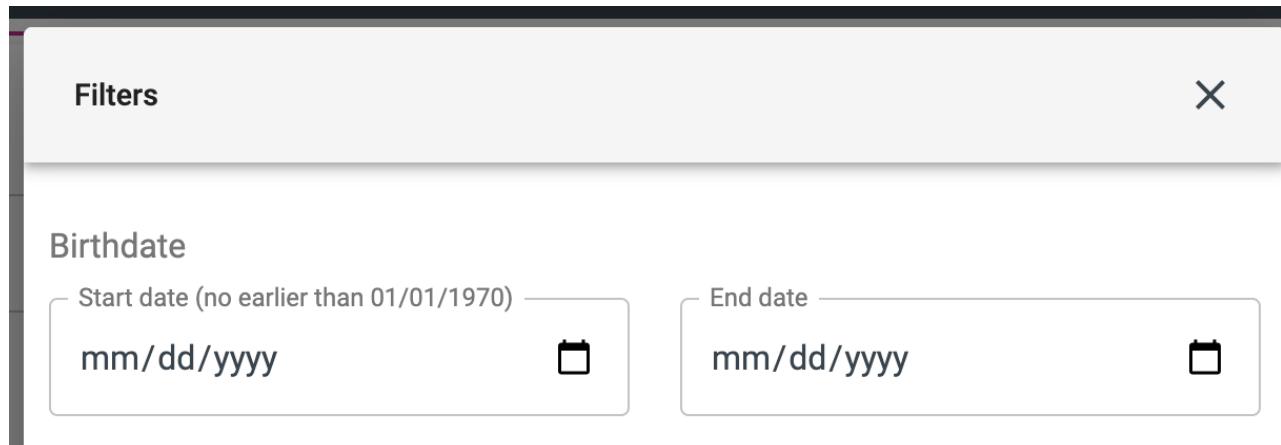
## 3.5 Filtering (cont.)

- Advanced filtering is specified in a filter dialog

```
"filters": {
 "keyword": {
 "fieldNames": [...]
 },
 "advanced": {
 "filterFields": [...]
 }
}
```

## 3.5 Filtering (cont.)

- Fields with the type of date, number, currency, and percent can be added to the filtering dialog
  - With two input fields, one for the minimum value and the other for the maximum value



```

"filters": {"advanced": {
 "filterFields": [{

 "fieldName": "Birthdate",
 "min": "$global.attr(minContactDob);",
 "minLabel": "Start date (no earlier than $global.attr(minContactDob, date, 'MM/dd/yyyy'))",
 "maxLabel": "End date",
 "minError": "Start date should not be earlier than $global.attr(minContactDob, date, 'MM/dd/yyyy');.",
 "minmaxError": "Start date should not be later than end date.",
 "maxminError": "End date should not be earlier than start date.",
 "fieldUi": {
 "settingsOptions": { "fieldFormat": "MM/dd/yyyy" },
 "format": { "options": { "datePattern": "MM/dd/yyyy" } }
 },
 }],
}

```

- min, max: specify the minimum or maximum limit
- minLabel, maxLabel: labels for the minimum and maximum inputs
- minError, maxError: error messages if the min value is less than the min limit or if the max value is greater than the max limit.
- minmaxError, maxminError: error messages if the min value is greater than the max value

```
"filters": {"advanced": {
 "filterFields": [
 {"fieldName": "Account.AnnualRevenue",
 "min": "$global.attr(minRevenue);", "max": "$global.attr(maxRevenue);",
 "minLabel": "Minimum revenue (no less than $global.attr(minRevenue, currency));",
 "maxLabel": "Maximum revenue (no greater than $global.attr(maxRevenue, currency));",
 "minError": "Invalid minimum revenue", "maxError": "Invalid maximum revenue",
 "minmaxError": "Minimum amount should not be larger than maximum amount.",
 "maxminError": "Maximum amount should not be smaller than minimum amount.",
 "fieldUi": {
 "settingsOptions": { "maximumFractionDigits": 0, "minimumFractionDigits": 0 },
 "format": {
 "options": {
 "numeral": true,
 "numeralThousandsGroupStyle": "thousand",
 "numeralDecimalScale": 0
 },
 "prefix": "$"
 }
 }
 },
 {"fieldName": "Birthdate",
 "fieldUi": {
 "settingsOptions": { "fieldFormat": "MM/dd/yyyy" },
 "format": { "options": { "datePattern": "MM/dd/yyyy" } }
 }
 }
]}
```

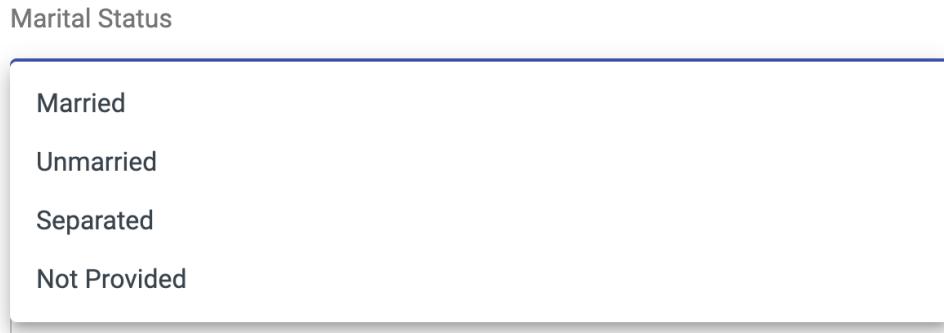
In fieldUi

- settingsOptions is for the chip display.
- format.options is for the dialog input.

## 3.5 Filtering (cont.)

- Fields with the type of picklist, radio buttons, and select buttons can be added to the filtering dialog
  - You can specify whether multiple option values can be selected. The record matches one value is kept.

```
"filters": {
 "advanced": {
 "filterFields": [{
 "fieldName":
 "clcommon__Marital_Status__c",
 "allowMultiple": true
 },
```



## 3.5 Filtering (cont.)

- Fields with the type of boolean, switch, and toggle can be added to the filtering dialog
  - As a picklist with the "not specified" text, the "true" text, and the "false" text

```
"filters": {
 "advanced": {
 "filterFields": [{
 "fieldName":
 "clcommon__Not_A_Citizen__c",
 "trueLabel": "true",
 "falseLabel": "false"
 } ,
```



## 3.5 Filtering (cont.)

- Chips can be cleared one by one or be cleared all together by clicking the Clear All link.
  - Chips and keyword filtering are combined.

Contacts

Filtered by

- Birthdate from **10/10/1970** X
- Marital Status **Married or Unmarried** X
- Not A Citizen **true** X

Clear All

Annual Revenue from **\$10,000 to \$100,000** X

Keyword  X

Full Name	Birthdate	Marital Status	Not A Citizen	Email	Account Name	Annual Revenue
Eric Smith	10/9/1999	Married	<input checked="" type="checkbox"/>		Bar	\$ 50,000.00

## 3.5 Filtering (cont.)

- Mobile UI (Mui only)

Filters X

Birthdate

Start date (no earlier than 01/01/1970)

mm/dd/yyyy

End date

mm/dd/yyyy

Marital Status

Not A Citizen

Annual Revenue

Minimum revenue (no less than \$10,000)

\$

Maximum revenue (no greater than \$1,000,000)

\$

CANCEL APPLY

## 3.5 Filtering (cont.)

- Mobile UI (Mui only)

The screenshot shows a mobile application interface for filtering contacts. At the top, it says "Contacts =". Below that, there are two buttons: one labeled "FOO" with "FOO" underneath, and another labeled "BAR" with "BAR" underneath. To the right of these buttons is a "Clear All" link. Below this section, there are four filter criteria listed in rounded rectangular boxes, each with a close button (an "X").

- Birthdate from 10/10/1970
- Marital Status Married or Unmarried
- Not A Citizen true
- Annual Revenue from \$10,000 to \$10...

Below the filters is a "Keyword" search bar containing the text "smith" with a clear button "X". At the bottom, there is a table with columns: Full Name, Birthdate, Marital Status, and Not A Citizen. The first row shows a contact named Eric Smith with a birthdate of 10/9/1999, marital status of Married, and a checked checkbox under "Not A Citizen".

Full Name	Birthdate	Marital Status	Not A Citizen
Eric Smith	10/9/1999	Married	<input checked="" type="checkbox"/>

# 3.5 Filtering (Mint)

**Filters**

**Birthdate**  
Start date (no earlier than 01/01/1970)

End date

**Marital Status**

Married  
 Unmarried  
 Separated  
 Not Provided

**Not A Citizen**

**Annual Revenue**  
Minimum revenue (no less than \$10,000)  
  
Maximum revenue (no greater than \$1,000,000)

**Cancel** **Apply**

**Filters**

**Birthdate**  
Start date (no earlier than 01/01/1970)

End date

**Marital Status**

Married  
 Unmarried  
 Separated  
 Not Provided

**Not A Citizen**

**Annual Revenue**  
Minimum revenue (no less than \$10,000)  
  
Maximum revenue (no greater than \$1,000,000)

**Cancel** **Apply**

```
"settings": {
 "filters": {
 "icon": "fa-filter"
 }
}
```

# 3.5 Filtering (Mint)

## Contacts

Filtered by

Birthdate from **10/10/1970** to **10/10/2001**

Marital Status **Married** or **Unmarried**

**Not A Citizen true**

**Clear All**

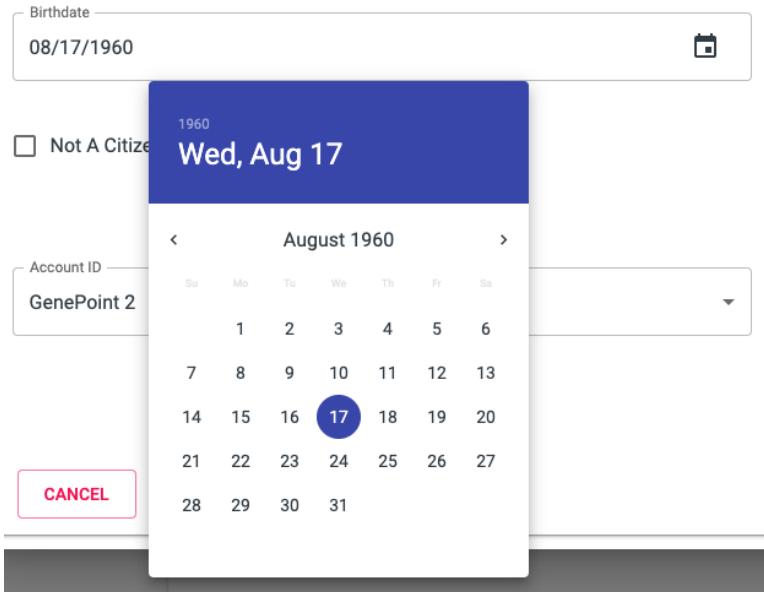
**Annual Revenue** from **\$10,000** to **\$100,000**

Keyword

Full Name	Birthdate	Marital Status	Not A Citizen	Email	Account Name	Annual Revenue
Eric Smith	10/9/1999	Married	<input checked="" type="checkbox"/> Not A Citizen		Bar	\$50,000.00

# 3.5 Mui Date Picker

- datepicker is a new fieldType for Mui, which is available across browsers, including Safari.



```
"fields": [{
 "id": "...",
 "fieldName": "Birthdate",
 "fieldUi": {
 "fieldType": "datepicker",
 "label": "Birthdate",
 "placeholderText": "mm/dd/yyyy",
 "format": {
 "options": {
 "datePattern": "MM/dd/yyyy"
 }
 }
 } ...
}
```

## 3.5 Mui Date Picker (cont.)

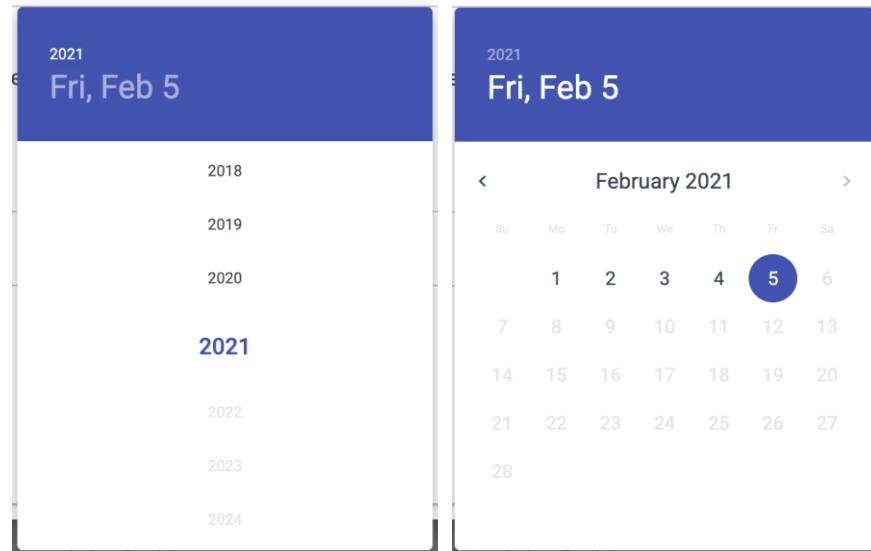
```
"format": {
 "options": {
 "datePattern": "MM/dd/yyyy",
 "disableFuture": true
 }
}
```

datePattern specifies the date format in the input field. datePattern uses **the date-fns format**

<https://date-fns.org/v2.16.1/docs/format>

datePattern can be set in the global settings

disableFuture and disablePast are supported by the datepicker.



## 3.5 Invalid Date Error Message

- Mui datepicker only

```
{
 "fieldName": "Birthdate",
 "fieldUi": {
 "fieldType": "datepicker",
 "messages": {
 "invalidDateText": "Invalid birthdate"
 }
 }
}
```



## 3.5 date-fns (no Moment!)

- From 3.5, date format uses the format specifications from date-fns
  - [Moment format is no longer supported.](https://date-fns.org/v2.16.1/docs/format)
- If the date format does not follow the date-fns format, the date is displayed as the Salesforce format of yyyy-mm-dd, e.g., 2020-05-10, **which can be used as an indicator** that your current date format configuration needs to be updated to the date-fns format.

# 3.5 Mui Overlay Title

- Overlay title can take portal variables

```
"overlay": [{
 "layoutId": "...",
 "name": "editContactOverlay",
 "title": "Contact: $global.attr(contactName) ;"
 "children": [{ ... }],
}]
```

The screenshot shows a modal dialog titled "Contact: Chris Smith". The dialog contains fields for First Name (Chris), Last Name (Smith), Birthdate (mm/dd/yyyy), and Marital Status (Unmarried). The Marital Status field includes a dropdown arrow. The dialog has a header bar with tabs: Full, Details, Marital, NOLA, Email, ACCOUNT, and Account Details. There is also a close button (X) in the top right corner.

Field	Value
First Name *	Chris
Last Name	Smith
Birthdate	mm/dd/yyyy
Marital Status	Unmarried

# 3.5 Mui Checkbox and Switch Enhancement

- Bold the checked / switched value



If you are not US citizen, check the checkbox



If you are not US citizen, check the checkbox



Not A Citizen

If you are not US citizen, turn on the switch



Not A Citizen

If you are not US citizen, turn on the switch

# 3.5 Mui Checkbox enhancement

- Checkbox can have both label and optionText with optionValue as true

```
{
 "fieldName": "clcommon__Not_A_Citizen__c", Not A Citizen
 "fieldUi": {
 "fieldType": "boolean",
 "label": "Not A Citizen",
 "tooltip": "If you are not US citizen,
check the checkbox",
 "selectOptions": [
 {
 "optionText": "Correct",
 "optionValue": true
 }]
 }
}
```

If you are not US citizen, check the checkbox

Not A Citizen

Correct

If you are not US citizen, check the checkbox

Not A Citizen

Correct

# 3.5 Mui Switch Enhancement

- Switch can have both label and two pieces of text at both end.
  - The left-end text is the optionText with the optionValue as false.
  - The right-end text is the optionText with the optionValue as true.

```
{ "fieldName": "clcommon__Not_A_Citizen__c",
 "fieldUi": {
 "fieldType": "switch",
 "label": "Not A Citizen",
 "tooltip": "If you are not US citizen, switch to Correct",
 "selectOptions": [{
 "optionText": "Wrong",
 "optionValue": false
 }, {
 "optionText": "Correct",
 "optionValue": true
 }]
 } }
```

Not A Citizen  
**Wrong**  **Correct**

If you are not US citizen, switch to Correct

**Not A Citizen**  
**Wrong**  **Correct**

If you are not US citizen, switch to Correct

## 3.5 Query with Post Process

```
"property": {
 "structureName": "contact_list_with_filtering",
 "queryProcess": {
 "post": "$ext.contactPostProcess"
 }
}
"structure": {
 "structureName": "contact_list_with_filtering",
 "tableName": "contact",
 "whereClause": {
 "freeForm": "id != null"
 },
 ...
}
 window.portalext = {};
In VF page window.portalext.contactPostProcess = (response) => {
 ...
 return response;
 }
```

## 3.5 Save with Pre/Post Process

```
"componentActions": [{
 "name": "save",
 "actionType": "save",
 "remoteActionProcess": {
 "pre": "$ext.saveContactPreProcess",
 "post": "$ext.saveContactPostProcess"
 },
 ...
}]
In VF page window.portalext = {};
window.portalext.saveContactPreProcess = (data) => {
 ...
 return data;
}
window.portalext.saveContactPostProcess = (response) => {
 ...
 return response;
}
```

## 3.5 Disclosure including Variables

- Both Mint and Mui
- Both confirmation message and description in the disclosure definition

I, **\$global.attr(userInfo).attr(firstName);  
\$global.attr(userInfo).attr(lastName);**, hereby acknowledge and agree to the statement.

Min Wu, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque egestas lectus euismod dui placerat, sit amet fringilla sapien gravida. Integer a gravida tellus, et accumsan leo. Nam vitae magna fringilla, scelerisque justo sed, bibendum sapien. Nulla sed malesuada metus, ut vestibulum eros. Nam vulputate non magna quis ultricies. Nullam nec lobortis nisi, faucibus malesuada lorem. Sed tincidunt congue posuere. Integer vitae nisi et sapien venenatis faucibus.

I, Min Wu, hereby acknowledge and agree to the statement.

## 3.5 Mui Disclosure Enhancement

- `navigate="open-disclosure"` as a special `navigate` value
  - Combined with the dialog configuration to display the disclosure description in a dialog.
  - Disclosure description is hidden by default.

Confirmation message: I hereby acknowledge and agree to [the statement](#)

```
"disclosures": [{
 "name": "SampleDisclosure",
 "dialog": {
 "title": "Sample Disclosure",
 "closeText": "I Agree"
 }
}]
```

I hereby acknowledge and agree to [the statement](#).

I hereby acknowledge and agree to [the statement](#).

### Sample Disclosure

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque egestas lectus euismod dui placerat, sit amet fringilla sapien gravida. Integer a gravida tellus, et accumsan leo. Nam vitae magna fringilla, scelerisque justo sed, bibendum sapien. Nulla sed malesuada metus, ut vestibulum eros. Nam vulputate non magna quis ultricies. Nullam nec lobortis nisi, faucibus malesuada lorem. Sed tincidunt congue posuere. Integer vitae nisi et sapien venenatis faucibus.

[I AGREE](#)

## 3.5 Mui Disclosure Enhancement

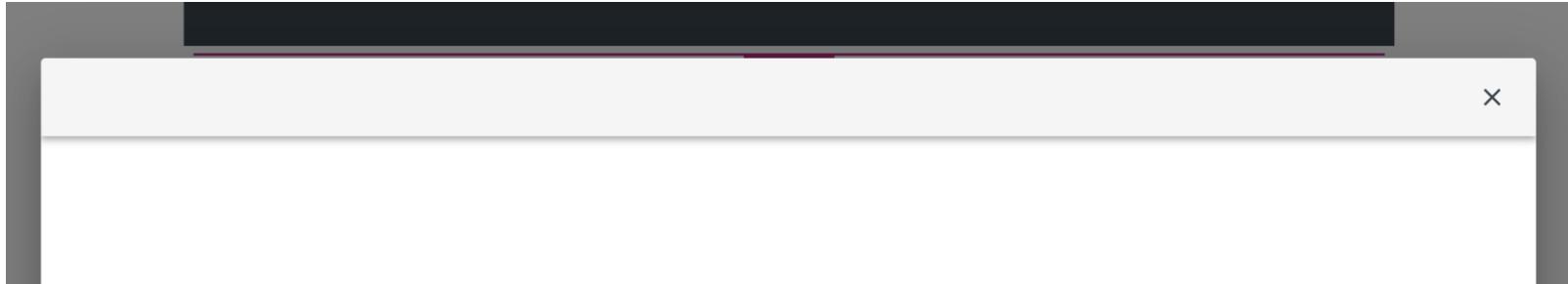
- Confirmation message can include other navigate values (different from `open-disclosure`), which can be handled by the navigation handler.

Confirmation message: I, \$global.attr(userInfo).attr(firstName); \$global.attr(userInfo).attr(lastName);, hereby acknowledge and agree to [the statement](#).

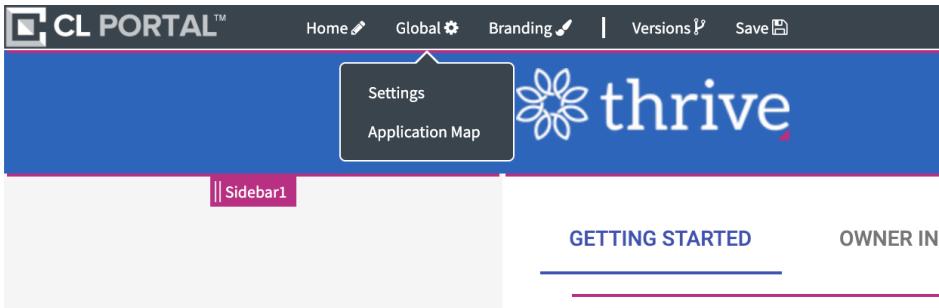
```
"generalHandlers": {
 "navigate": {
 "open-overlay": {
 "updates": [{
 "name": "$nav.overlay.add;",
 "value": "DisclosureOverlay"
 }]
 }
 }
}
```

Min Wu, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque egestas lectus euismod dui placerat, sit amet fringilla sapien gravida. Integer a gravida tellus, et accumsan leo. Nam vitae magna fringilla, scelerisque justo sed, bibendum sapien. Nulla sed malesuada metus, ut vestibulum eros. Nam vulputate non magna quis ultricies. Nullam nec lobortis nisi, faucibus malesuada lorem. Sed tincidunt congue posuere. Integer vitae nisi et sapien venenatis faucibus.

I, Min Wu, hereby acknowledge and agree to [the statement](#).



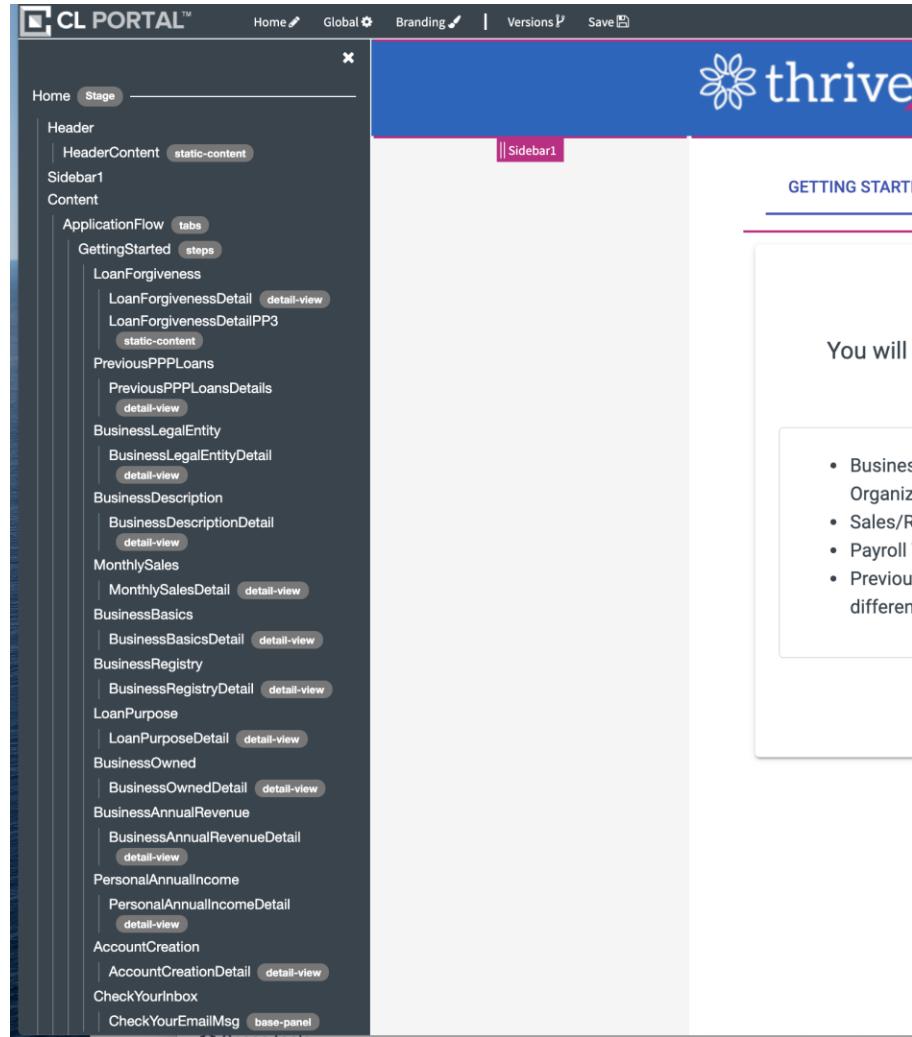
# 3.6 Application Map

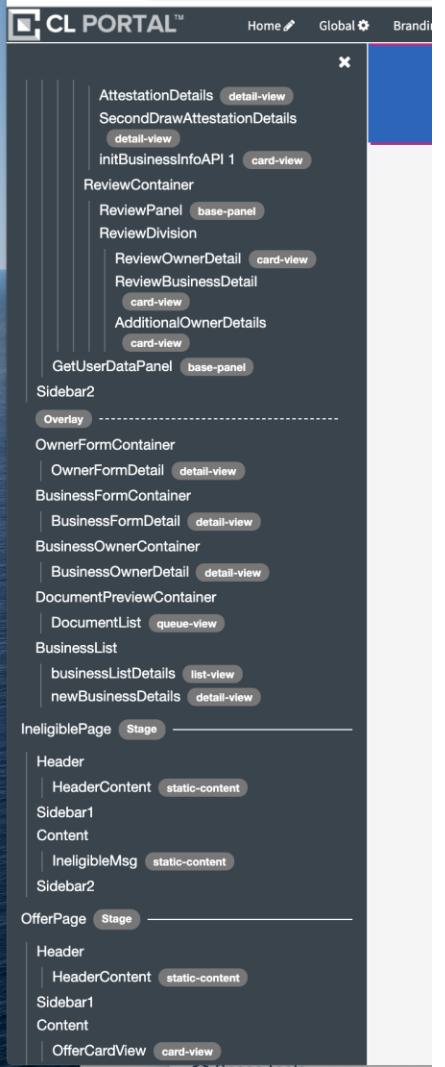


Clicking Application Map opens a left-hand sidebar which includes the stage and actor structure of the portal application

**Need to update the builder configuration using**

```
node builder-loader.js username
password token=security_token
```





# 3.6 Application Map

- Actors are grouped by stages.
- If an actor is in a certain stage, clicking the actor goes to the corresponding stage.
- If an actor is in a certain tab/step, clicking the actor opens the corresponding tab/step.
  - Note that you may need to scroll to see the tab/step
- If an actor is in a certain overlay, clicking the actor opens the corresponding overlay.
- Data in actors may not be properly loaded because in the real application flow, the data in an actor may depend on another actor.

## 3.6 Mui Popover

- Popover can be triggered by the action with the `actionType` of context.
- Any overlay can be triggered as popover.

```
"componentActions": [{
 "name": "...",
 "actionType": "context",
 "popup": {
 "direction": "bottom",
 "anchor": "right",
 "overlay": "PopoverContainer"
 }
}]
```

```
"overlays": [{
 "name": "PopoverContainer",
 "children": [...],
 "width": "300px",
 "maxHeight": "200px"
}]
```

|| PopoverPanel

SHOW POPOVER

|| AccountList

Foo



Foo



Foo



foo

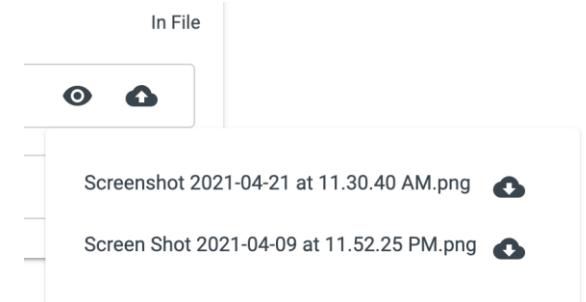
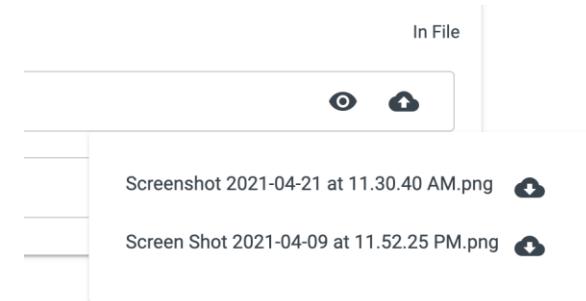
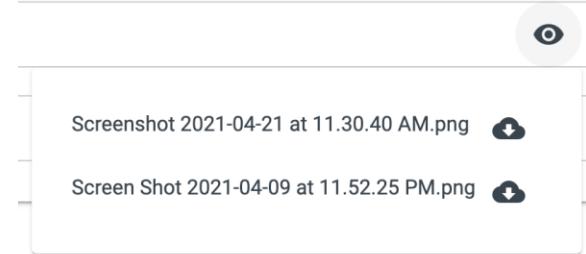


## 3.6 Popover Positions

```
"componentActions": [{
 "popup": {
 "direction": "bottom",
 "anchor": "right",
 "overlay": "DocumentPreviewContainer"
 }, ...
},]

"popup": {
 "direction": "bottom",
 "overlay": "DocumentPreviewContainer"
}

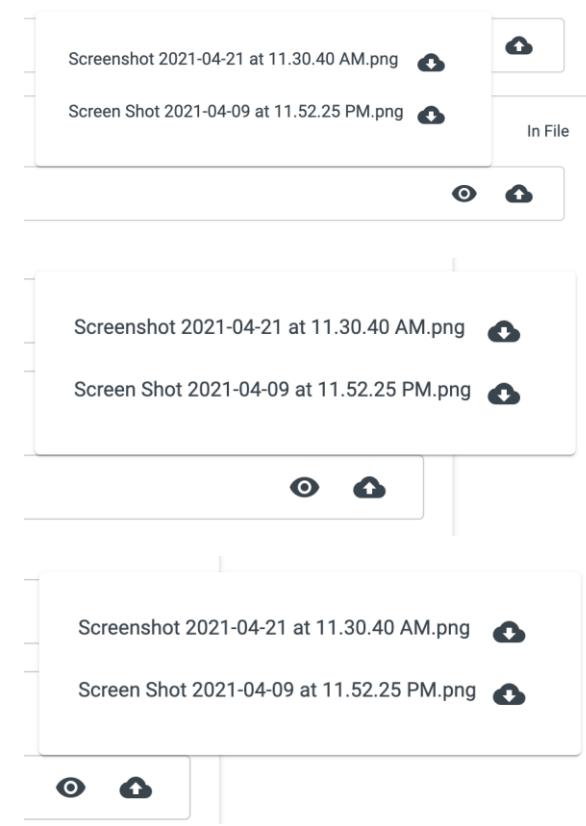
"popup": {
 "direction": "bottom",
 "anchor": "left",
 "overlay": "DocumentPreviewContainer"
}
```



## 3.6 Popover Positions (cont.)

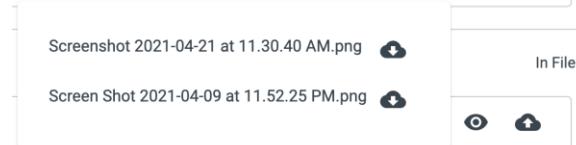
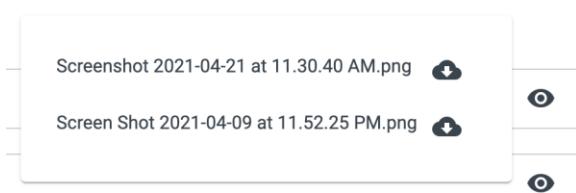
```
"componentActions": [{
 "popup": {
 "direction": "top",
 "anchor": "right",
 "overlay": "DocumentPreviewContainer"
 }, ...
},]
"popup": {
 "direction": "top",
 "overlay": "DocumentPreviewContainer"
}

"popup": {
 "direction": "top",
 "anchor": "left",
 "overlay": "DocumentPreviewContainer"
}
```



## 3.6 Popover Positions (cont.)

```
"componentActions": [{
 "popup": {
 "direction": "left",
 "anchor": "top",
 "overlay": "DocumentPreviewContainer"
 }, ...
},]
"popup": {
 "direction": "left",
 "overlay": "DocumentPreviewContainer"
}
"popup": {
 "direction": "left",
 "anchor": "bottom",
 "overlay": "DocumentPreviewContainer"
}
```



## 3.6 Popover Positions (cont.)

```
"componentActions": [{
 "popup": {
 "direction": "right",
 "anchor": "top",
 "overlay": "DocumentPreviewContainer"
 }, ...
},]
"popup": {
 "direction": "right",
 "overlay": "DocumentPreviewContainer"
}

"popup": {
 "direction": "right",
 "anchor": "bottom",
 "overlay": "DocumentPreviewContainer"
}
```



# 3.6 Popover Dimension

- Popover dimension includes
  - **width, minWidth, and maxWidth**
    - **Width or maxWidth is required.**
  - **height, minHeight, and maxHeight**
    - **height or maxHeight is required.**

```
"overlays": [
 {
 "name": "...",
 "children": [...],
 "width "maxHeight }
]
```

## 3.6 Overlay Dimension

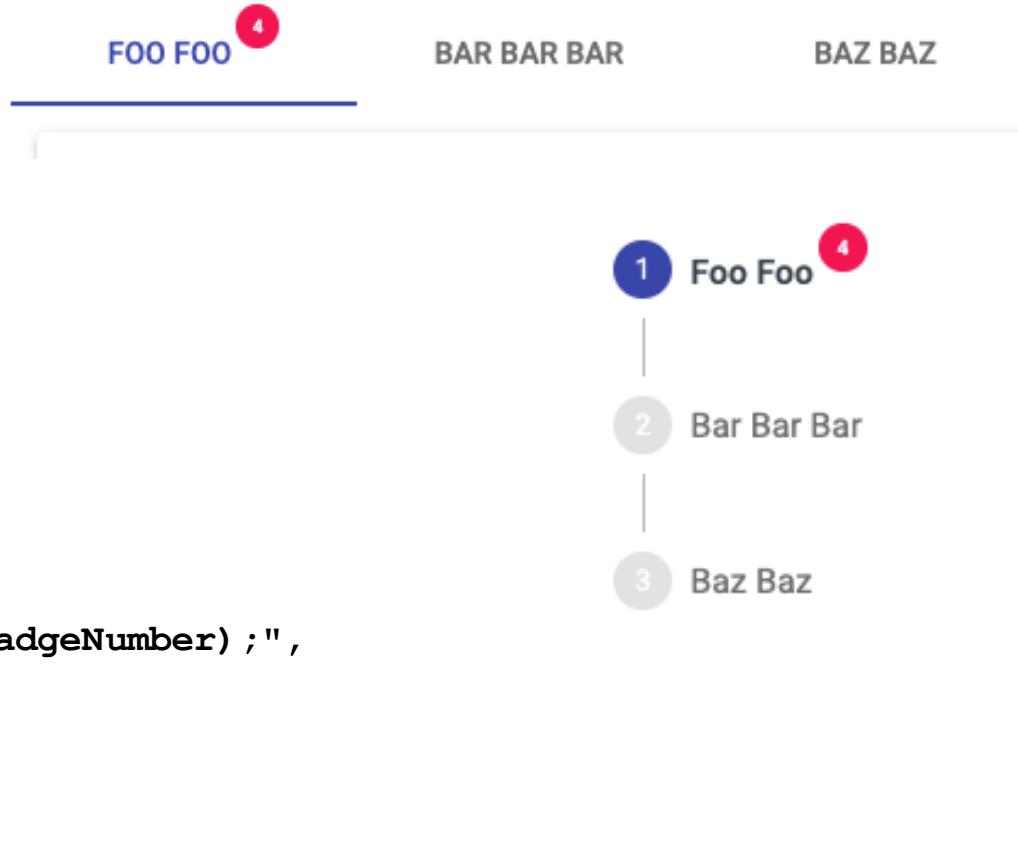
- Dimension works for layout as well (both Mui and Mint)
  - **width (recommended), minWidth, and maxWidth**
  - **height, minHeight, and maxHeight**

```
"overlays": [
 {
 "name": "...",
 "children": [...],
 "width": "400px",
 "maxHeight": "300px"
 }
]
```

## 3.6 Mui Badge

- Badge can be added to
  - the titles of tabs/steps and
  - the actions

```
"appearance": {
 "titles": [
 {
 "title": "Foo Foo",
 "badge": {
 "content": "$global.attr(badgeNumber) ;",
 "render": true
 }
 }, {...}
]
}
```



## 3.6 Mui Badge (cont.)

```
"componentActions": [
 {
 "name": "FooFooButton1",
 "actionType": "normal",
 "appearance": {
 "label": "Button One",
 "style": "primary",
 "position": "top-right",
 "badge": {
 "content": 40,
 "render": true
 }
 }
 }, ...
]
```



## 3.6 Mui Badge (cont.)

- Mui badge supports the following configuration attributes from material-ui badge (<https://material-ui.com/components/badges/>)
  - max
  - showZero
  - variant: can be "dot" or "standard"
  - overlap: can be "circle" or "rectangle"
  - anchorOrigin
  - color: "secondary" (default), "primary", or "error"

## 3.6 Mui Badge (cont.)

```
"badge": {
 "content": "$global.attr(largeNumber);",
 "max": 99
}
"global": {
 "largeNumber": 170
}
```

```
"badge": {
 "variant": "dot",
 "color": "primary",
 "anchorOrigin": {
 "vertical": "bottom",
 "horizontal": "left"
 }
}
```



## 3.6 Upload Large Document

- Document that is greater than 25 MB can be uploaded as content document
  - If `maxSize` is not specified, document size is limited to 25 MB.

```
"property": {
 "maxSize "uploadType": "document"
}
```

## 3.7- Titanium

## 3.7: Hide close button in overlay (CP-1992)

- We can hide the close button in overlay using 'hideCloseButton'
- In overlay, set "hideCloseButton" to true to hide the close button.

```
"overlay": [{
 "name": "preview",
 "title": "...",
 "hideCloseButton": true,
 ...
}
```

# Todo List

Not Fully Implemented Yet!

## 3.6 Popover ClickAway

- (recommended) By default, when a popover is open by clicking an action, a user has to click the action again to close the popover.
- By configuring a popover `clickAway: true`, the popover will be closed by clicking somewhere outside it.
  - Known issue: clicking any button won't close the popover because of `stopPropagation`.

```
"componentActions": [{
 "popup": {
 "direction": "left",
 "anchor": "top",
 "overlay": "PopoverContainer",
 "clickAway": true
 }, ...
}
```

### 3.3 Multi-Branding + Logo

- `styleId` can be added to the "app" node in the VF page

```
<div id="app" pageId="BorrowerPortal1"
styleId="investors"></div>
```

so that different branding configuration (from the default branding named as the `pageId`) can be applied to the portal application.

## 3.3 Multi-Branding + Logo (cont.)

- Add two special names to the `backgroundImages` configuration
  - `$mainLogo`: whose static resource name will be `<styleId>_mainLogo`
  - `$secondaryLogo`: whose static resource name will be `<styleId>_secondaryLogo`

Default branding without styleId <div id="app" pageId="BorrowerPortal1"></div>

```
{ "backgroundImages": [{
 ...
 "name": "$mainLogo",
}] }
```

With the static resource

BorrowerPortal1\_mainLogo

thrive

WELCOME SBA60 ▾

Usage Logs Usage Documents Analytics

### Checkpoint 1

Welcome to the SBA Portal! We use "checkpoints" as opportunities to monitor how you are using the proceeds of your SBA Paycheck Protection loan. Over the next few weeks, we will ask you to submit information such as: How many employees do you have? Or how much have you spent on payroll? This information is used to verify permissible (and forgivable) use of funds. Please complete the fields below and click Submit, and then click the Usage Documents tab to upload required documents.

Please enter your current Number of Employees (as of today)  
0

Please enter the amount spent during the previous XX weeks on

Payroll Usage \$ 0	Group Healthcare Usage \$ 0
Rent Usage \$ 0	Mortgage Interest Usage \$ 0
Utilities Usage \$ 0	

Total Loan Amount Used : \$0.00

SUBMIT USAGE

**Messages**

⚠ Failed to load ⌂

**Why do we need this Data?**

In order to have your loan forgiven, you must :

- Use 75% or more of the proceeds on Payroll Costs, and
- Use up to 25% of the proceeds on other permissible uses(such as Rent, Utilities, Group Benefits and Mortgage Interest)

We use this documentation to verify permissible use of the funds you have received on your Payment Protection Program loan.

```
<div id="app" pageId="BorrowerPortal1" styleId="investors"></div>
```

The screenshot shows the SBA Portal application interface. At the top, there is a green header bar with the 'investorsBank' logo on the left and a 'WELCOME SBA60' dropdown on the right. A red arrow points from the text 'With the static resource investors\_mainLogo' to the 'investorsBank' logo. Below the header, there are three tabs: 'Usage Logs' (which is active), 'Usage Documents', and 'Analytics'. The main content area is titled 'Checkpoint 1'. It contains a welcome message about checkpoints and instructions to complete fields. There are five input fields for different types of usage: 'Payroll Usage' (\$0), 'Group Healthcare Usage' (\$0), 'Rent Usage' (\$0), 'Mortgage Interest Usage' (\$0), and 'Utilities Usage' (\$0). A note below the fields states 'Total Loan Amount Used : \$0.00'. At the bottom right is a green 'SUBMIT USAGE' button. To the right of the main content, there is a 'Messages' box containing a warning message: '⚠ Failed to load ⌂'.

```
{ "backgroundImages": [{
 ...
 "name": "$mainLogo",
}] }
```

With the static resource  
investors\_mainLogo

# Nested Component Structure

```
{ "structureName": "document_category",
 "tableName": "clcommon_Document_Category_c",
 "sections": [{
 "fields": [{
 "id": "...",
 "fieldName": "clcommon_Category_Name_c",
 "fieldUi": { "fieldType": "string", "label": "Category Name" }
 }, {
 "id": "...",
 "fieldName": "Attachment_Parent_Id_c",
 "fieldUi": { "fieldType": "string", "label": "Attachment Parent Id" }
 }, {
 "id": "...",
 "fieldName": "clcommon_Attachment_Id_c",
 "fieldUi": { "fieldType": "string", "label": "Attachment ID" }
 }, {
 "id": "...",
 "relationType": "childrenRelation",
 "relationName": "clcommon_DocCat_Attachment_Associations_r",
 "section": "Attachment"
 }
]
}
```

# Nested Component Structure (cont.)

```
{ "structureName": "attachment_view",
 "tableName": "clcommon_Category_Attachment_Association_c",
 "whereClause": {"freeForm": "clcommon_Document_Category_c = '$ParentId'"} }
 "sections": [{
 "fields": [{
 "id": "...",
 "fieldName": "Id",
 "fieldUi": { "fieldType": "string", "label": "Record ID" }
 }, {
 "id": "...",
 "fieldName": "Name",
 "fieldUi": { "fieldType": "string", "label": "Name" }
 }, {
 "id": "...",
 "relationType": "childrenRelation",
 "relationName": "clcommon_DocCat_Attachment_Associations_r",
 "relationStructureName": "attachment_view"
 }, {
 "id": "...",
 "label": "Attachment"
 }]
 }] }
```