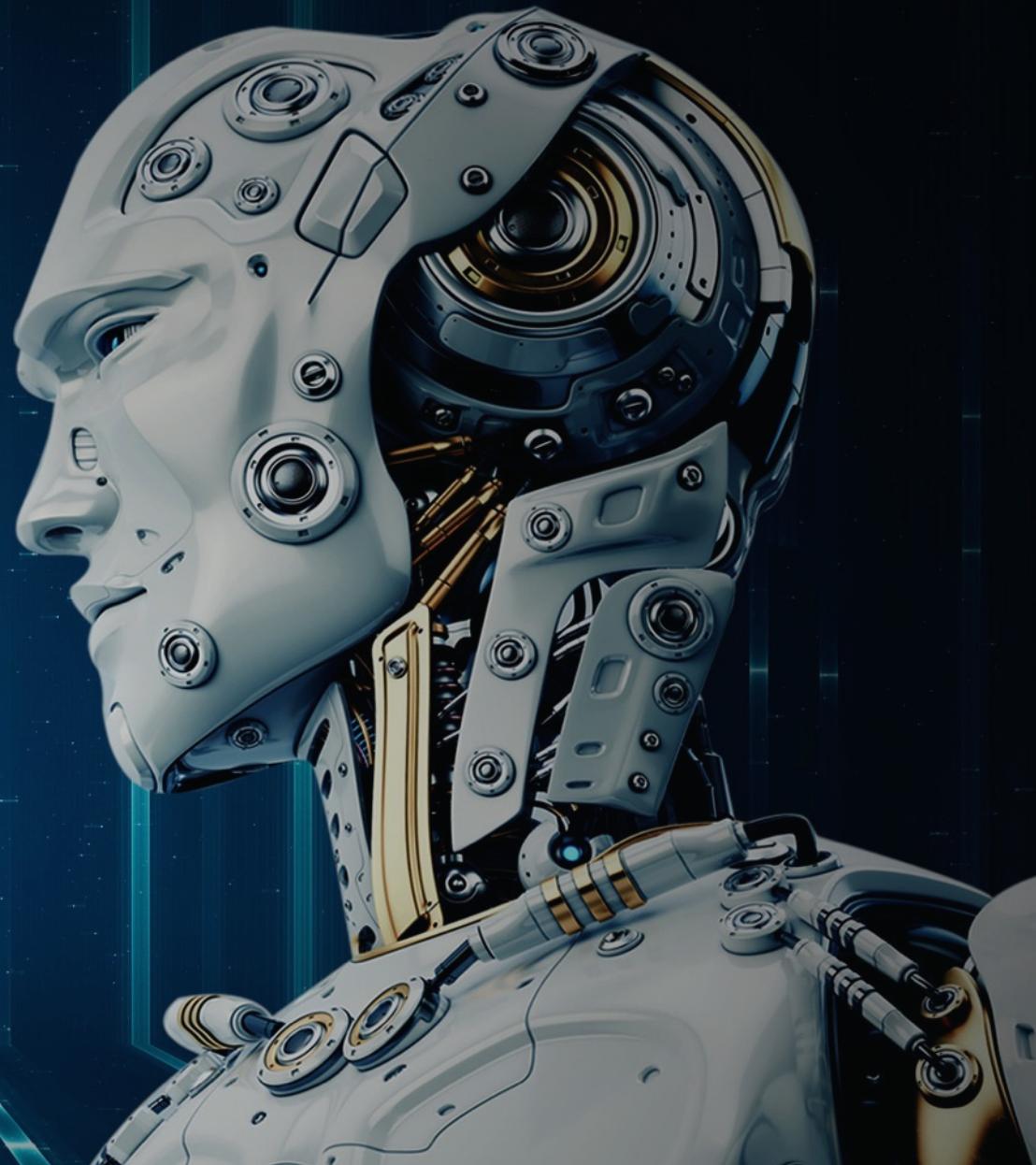


ORACLE®

Oracle Intelligent Bots Advanced Training

Instant Apps

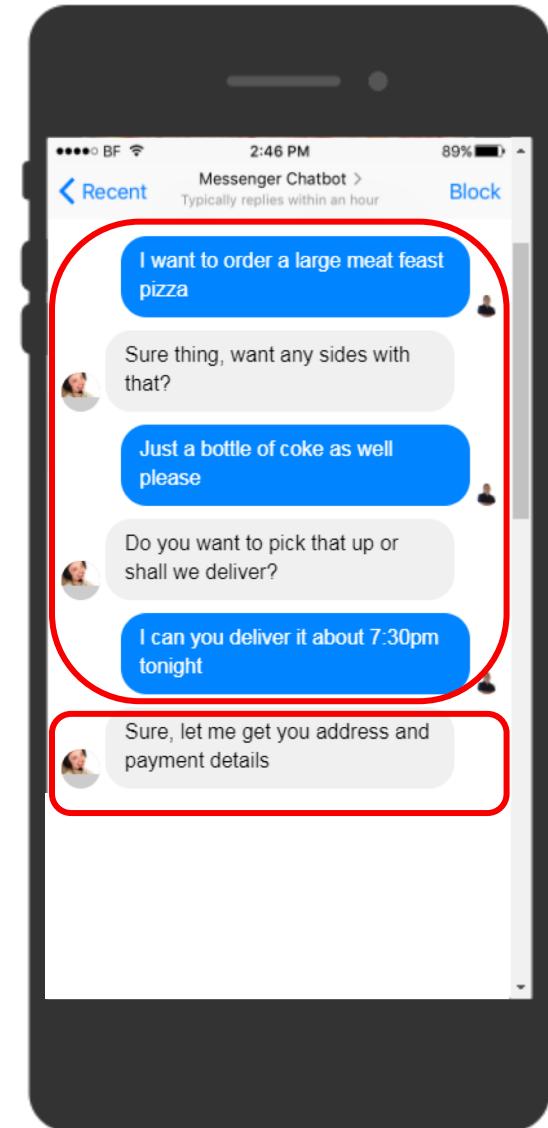


Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

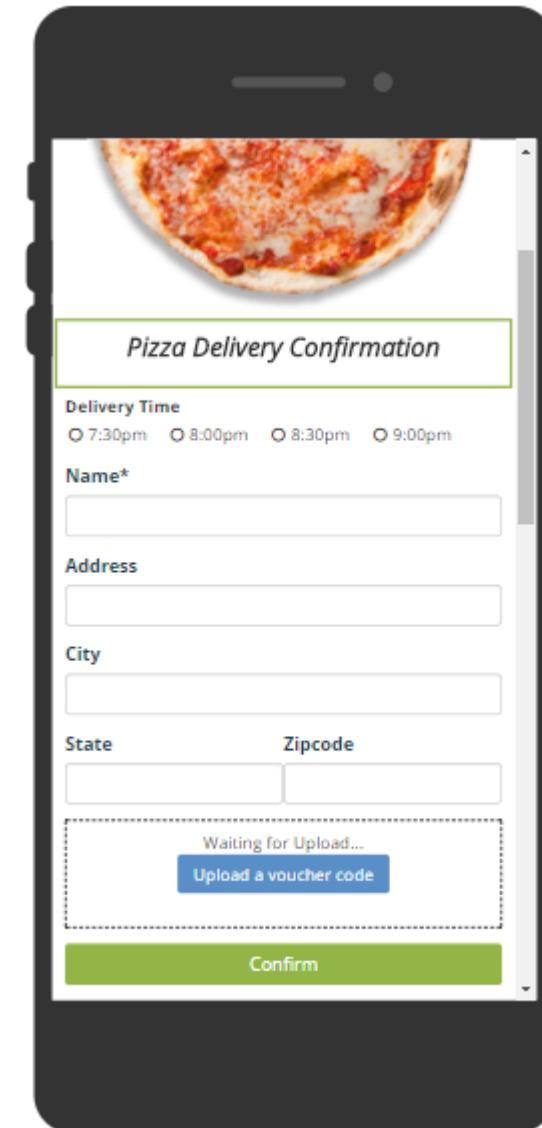
A Typical Conversation

- Natural language conversation
 - Unstructured
 - Free flowing
 - “Conversational”
- But not all conversation
 - Credit card details
 - Delivery address
 - Scanning a voucher/coupon
 - Viewing delivery progress on a map

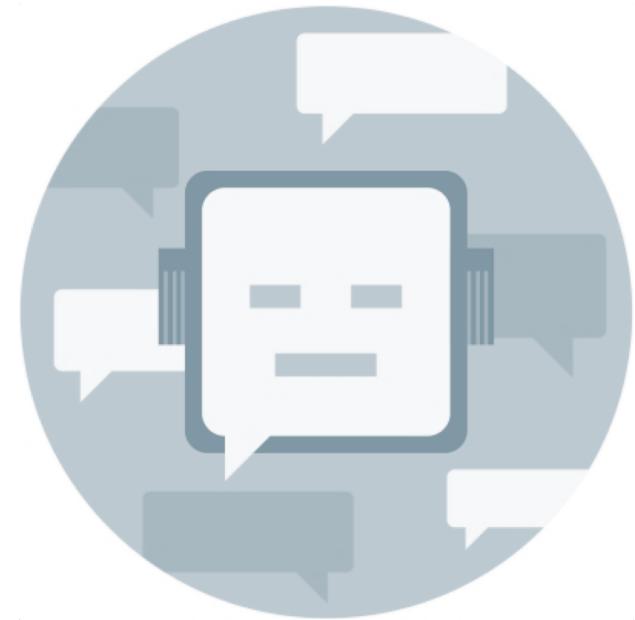


A Typical Conversation

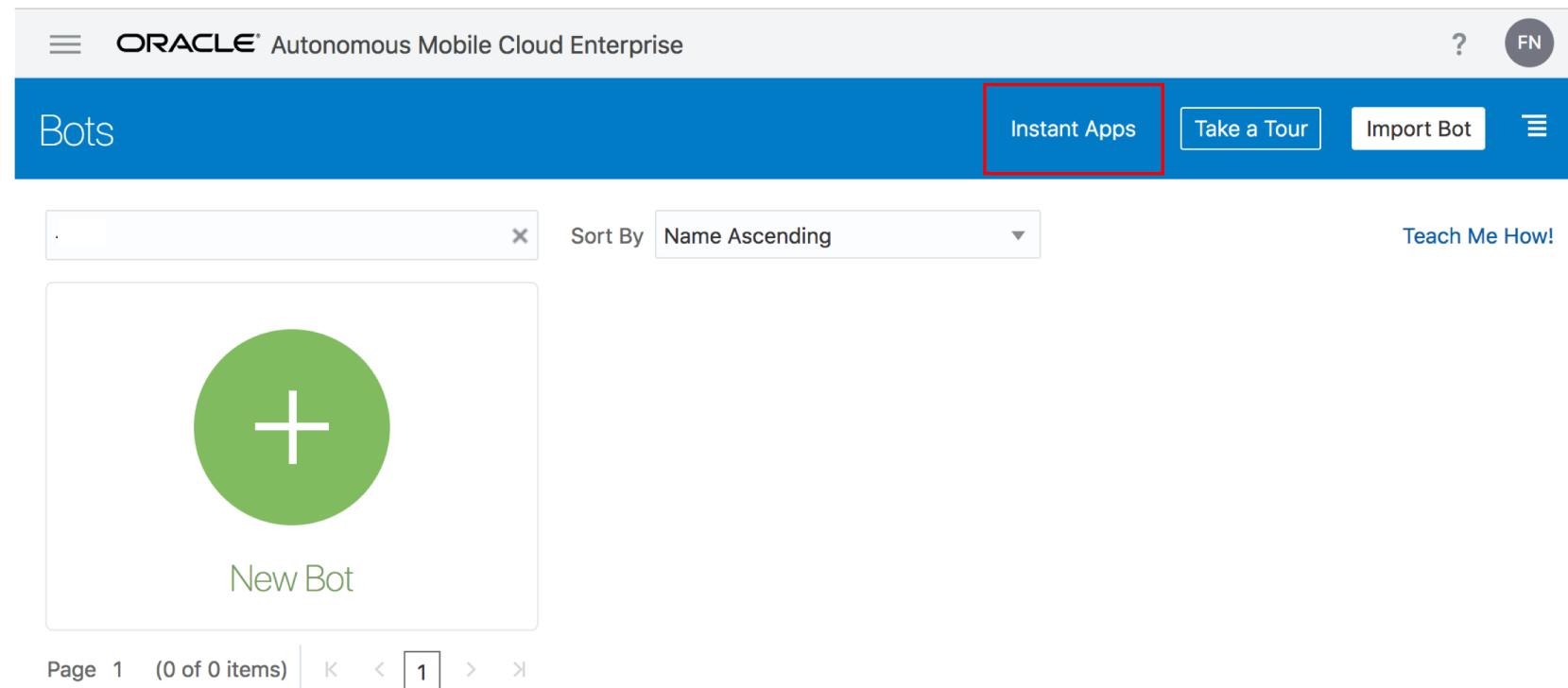
- Some parts of a conversation are better handled in a structured way
 - User knows exactly what information is required
 - Discrete choices
 - Faster input
 - Immediate validation of input
 - Information capture
 - Upload image
 - Signature
 - Scan barcode



**Instant Apps allow you to shorten
bot conversations**



Creating Instant Apps



The screenshot shows the Oracle Autonomous Mobile Cloud Enterprise interface. At the top, the navigation bar includes the Oracle logo and the text "Autonomous Mobile Cloud Enterprise". On the right side of the bar are a help icon (?), a user profile icon (FN), and a three-dot menu icon.

The main header is "Bots". Below it, there are several buttons: "Instant Apps" (which is highlighted with a red box), "Take a Tour", "Import Bot", and another three-dot menu icon. To the left of these buttons is a search bar with a placeholder ". . ." and an "X" button.

Below the header, there is a sorting section labeled "Sort By" with a dropdown menu set to "Name Ascending". To the right of this is a "Teach Me How!" link. The main content area features a large green circle with a white plus sign, labeled "New Bot".

At the bottom, there is a pagination section showing "Page 1 (0 of 0 items)" followed by a page number input field containing the number "1".

Starter Templates

Templates

Select a template from which to build your Instant app -- or you can create your own from scratch.



Blank

Create your own Instant App from scratch



Content Delivery

Rich media experiences of photos, PDFs, video



Customer Survey

Get feedback from your customers and collect email to follow up on their experience



Dispute Transaction

Present transaction information, discover dispute reason, upload receipt photo, update contact information



Internal Acknowledgement

Collect acknowledgments and signatures from employees



Lead Form

Collect lead information and schedule demos



Refund Request

Request information about a broken item for a refund



Simple Form

Collect data in fields, upload photo, ask structured questions



Simple Tech Support

Provide troubleshooting steps and scan barcodes to help identify product information



Weather

Call an External Web API to retrieve real-time weather data for Seattle and display in a chart

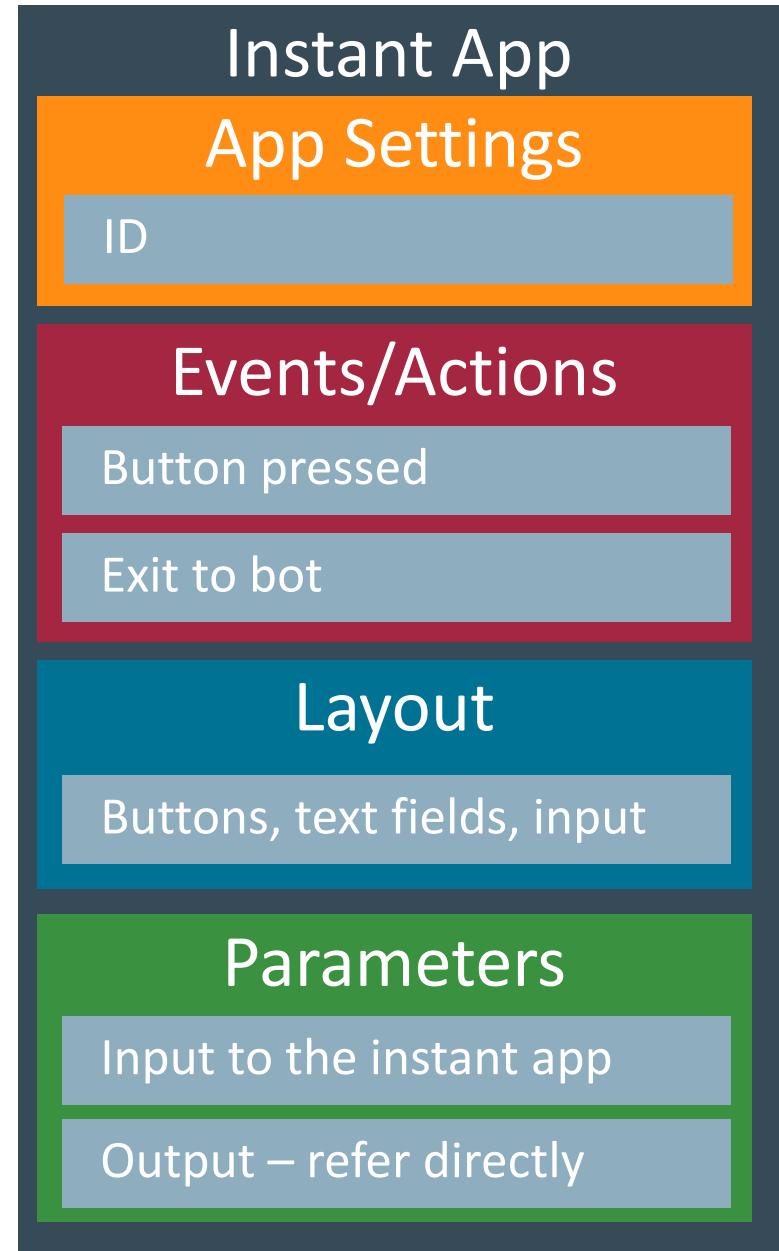


Work Order

Send work order information and collect pictures & confirmation on site

Anatomy of an Instant App

- Four main areas
- Application configuration
 - id, display image
- Ability to create events for UI components
 - value change, button press, app-launch etc.
- Ability to create layouts and panels
 - option to navigate between panels
- Definition of in- and output parameters



Anatomy of an Instant Apps

24hrsflowersPayment

App Settings

Layout

Events and Actions

Parameters

Localizations

Name: 24hrsflowersPayment

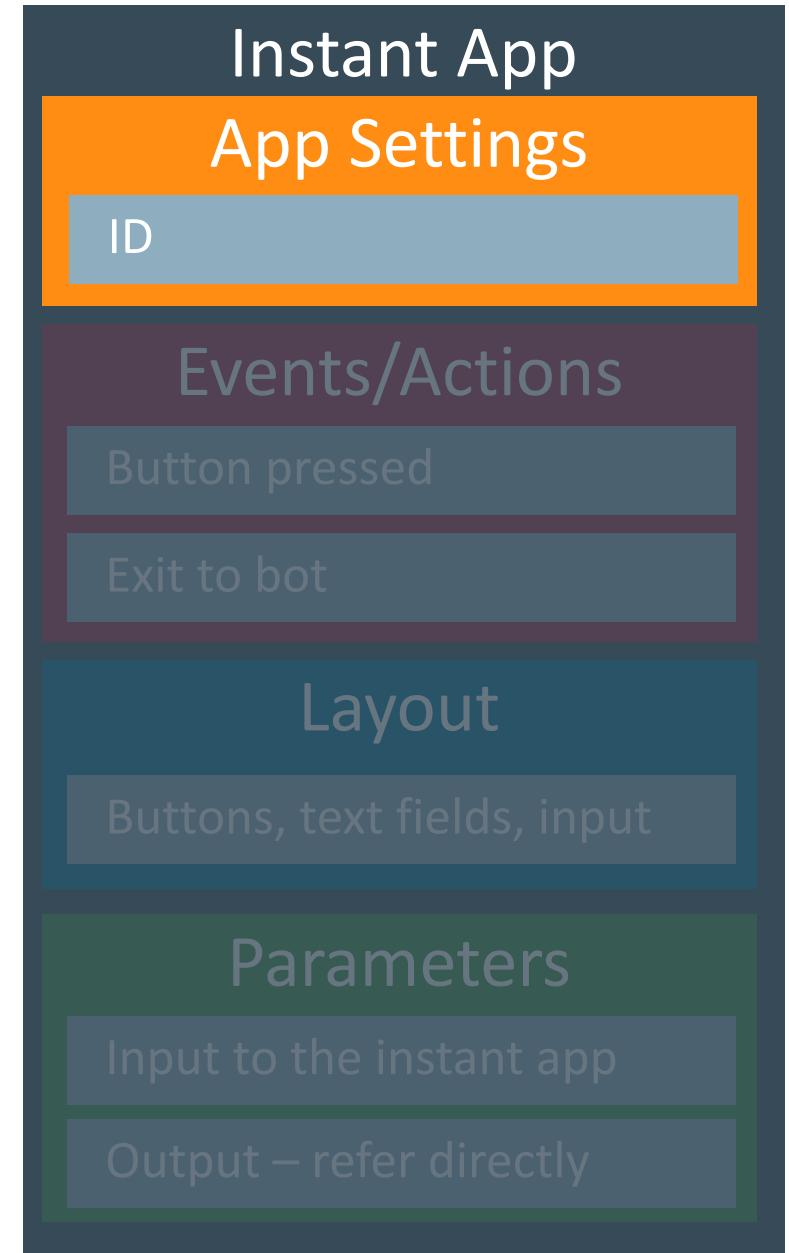
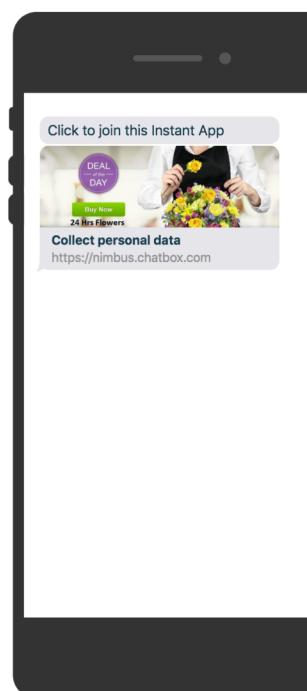
ID: 24hrsflowersPayment

Icon: 

Internal Description: Payment Integration

Invite Message: Click to join this Instant App {link}

Preview Test JSON



Anatomy of an Instant Apps

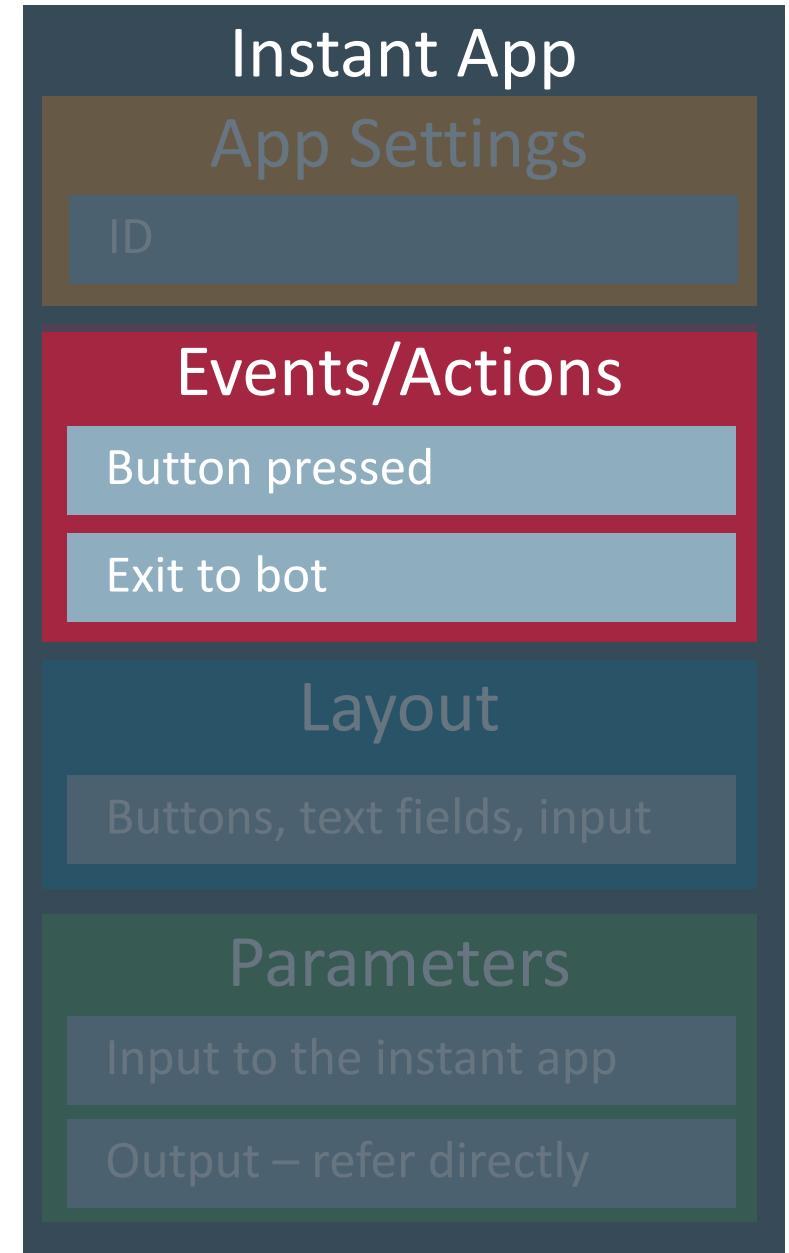
< 24hrsflowersPayment

The screenshot shows the configuration interface for the '24hrsflowersPayment' instant app. On the left, a sidebar lists navigation options: App Settings, Layout, **Events and Actions** (selected), Parameters, and Localizations. The main area is divided into two sections: 'App Events' and 'FistPage Events'.
App Events:

- App Sent
- Customer Connected
- Customer Disconnected
- App Locked
- App Unlocked (highlighted)

FistPage Events:

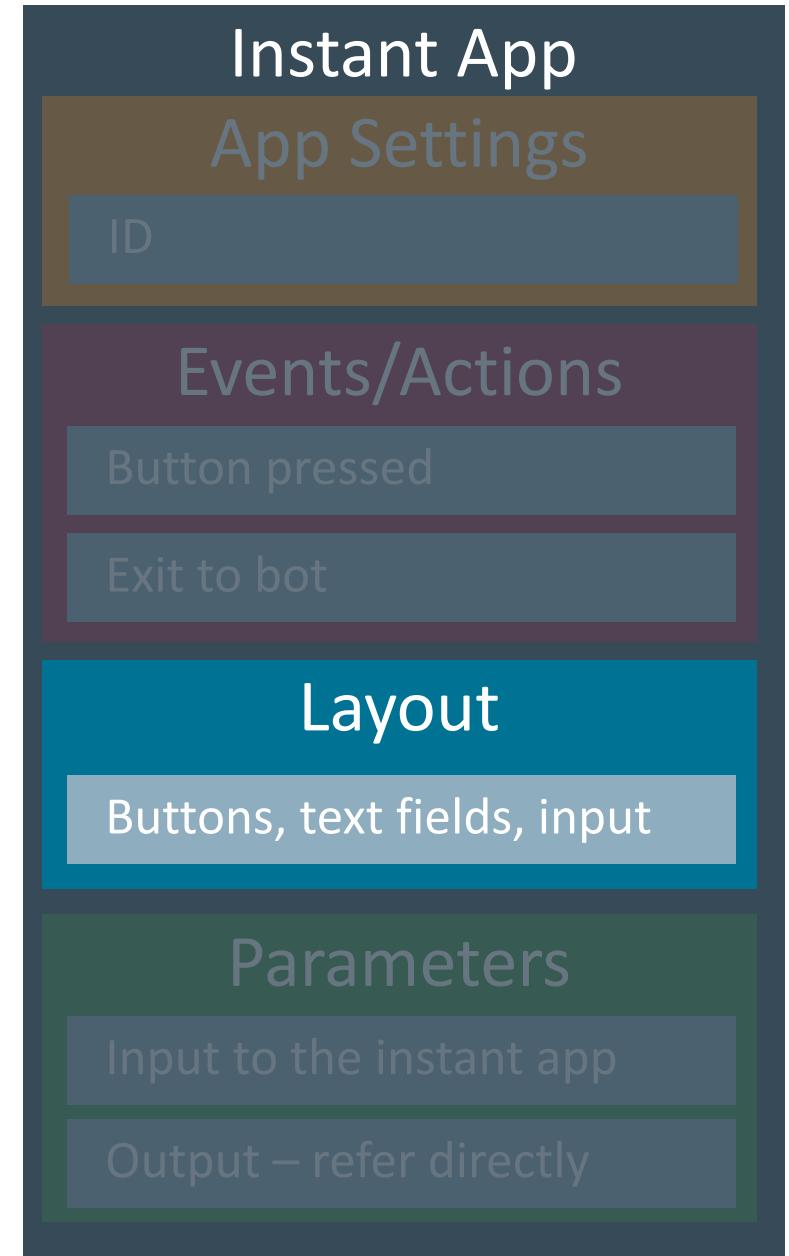
- select_card_type Changed
- input_name Changed
- input_card_number Changed
- input_exp_date Changed
- input_verifi_code Changed
- reset Pressed
- Validate Pressed
- button_submit Pressed



Anatomy of an Instant Apps

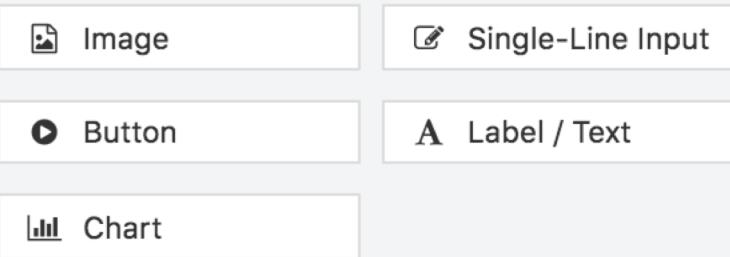
◀ 24hrsflowersPayment

The screenshot shows the Oracle Instant Apps configuration interface. The left sidebar lists various UI elements: image_header_1, label_secure, label_cc, messageKO, select_card_type, input_name, input_card_number, input_exp_date, input_verifi_code, messageOK, reset, Validate, button_submit, and + Add Element. The main area has tabs for 'Configure' and 'Validation'. Under 'Configure', there's a 'Pane Settings' section with a 'Pane ID' set to 'FirstPage'. Below it is an 'Add Element' section with categories: 'Common Elements' (Image, Single-Line Input, Button, Label / Text, Chart), 'Text Inputs' (Single-Line Input, Multi-Line Input, Email, Number, Phone, Website Address, Rich Text), and 'Choice Inputs' (Checkbox, Radio Buttons, Pick List). The 'Configure' tab is selected.

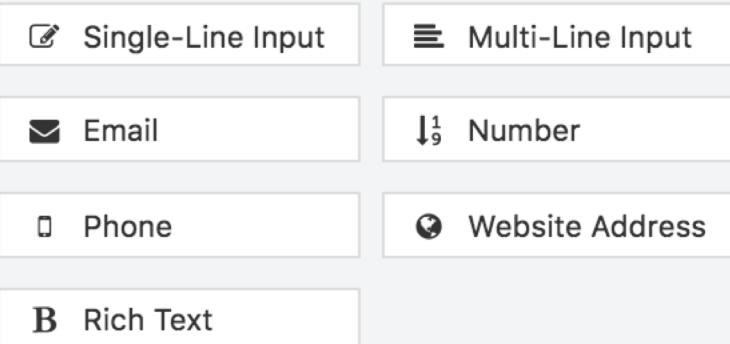


UI Component Overview

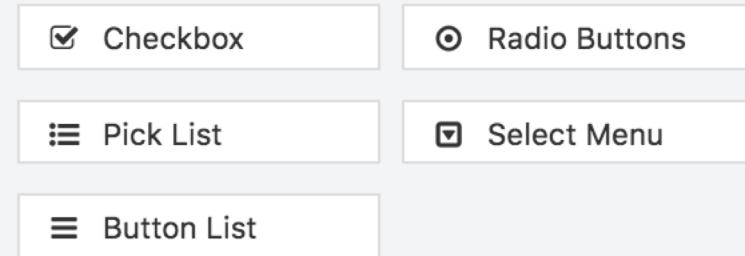
Common Elements



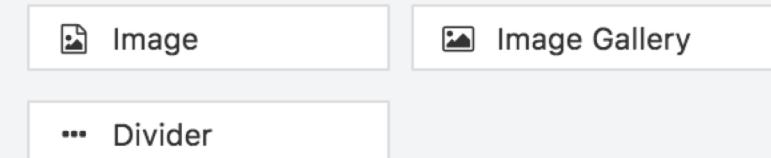
Text Inputs



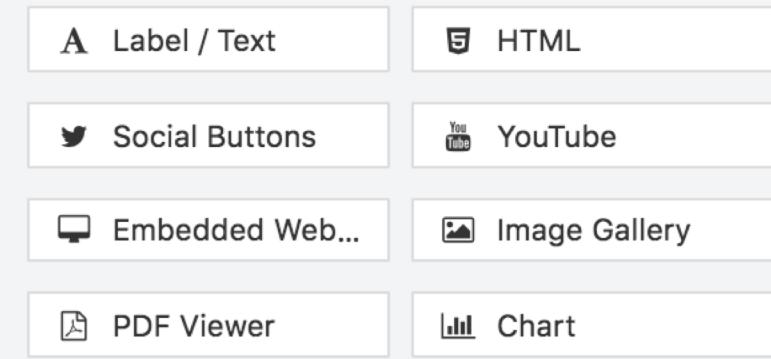
Choice Inputs



Images & Layout



Content



Anatomy of an Instant Apps

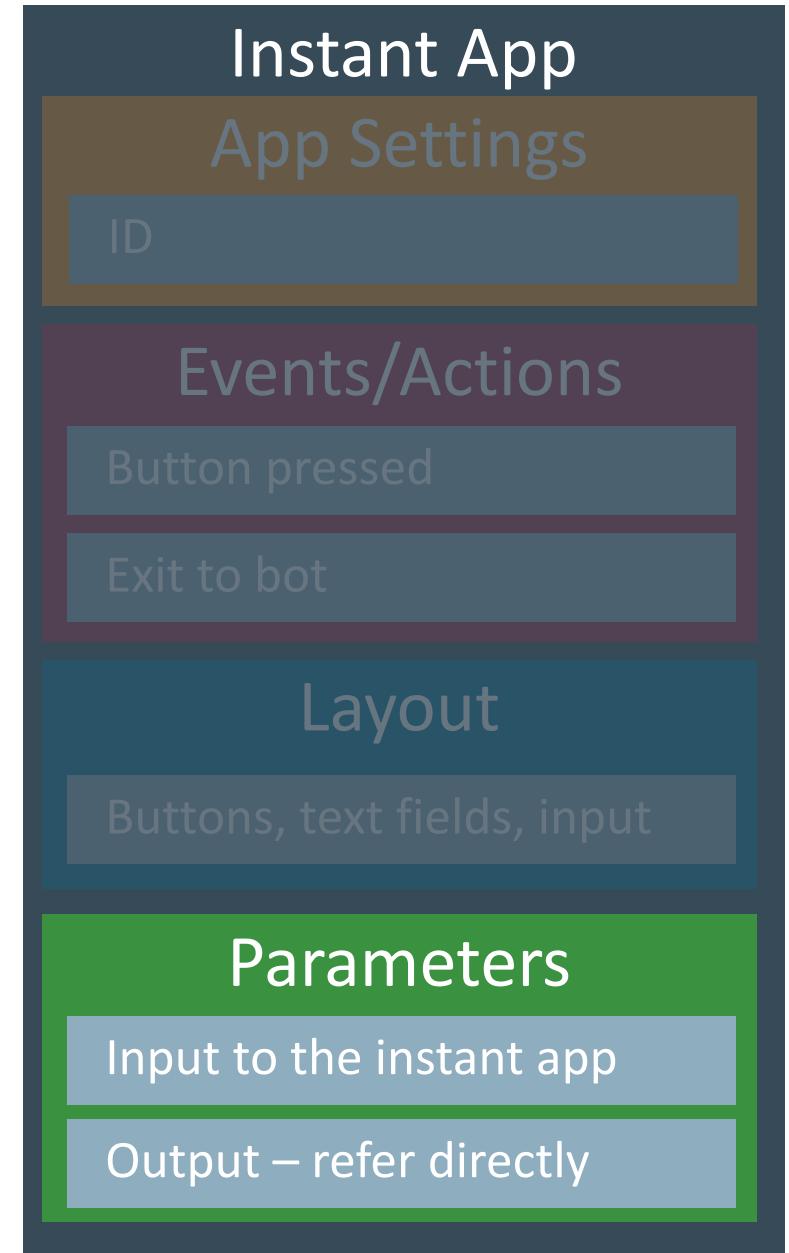
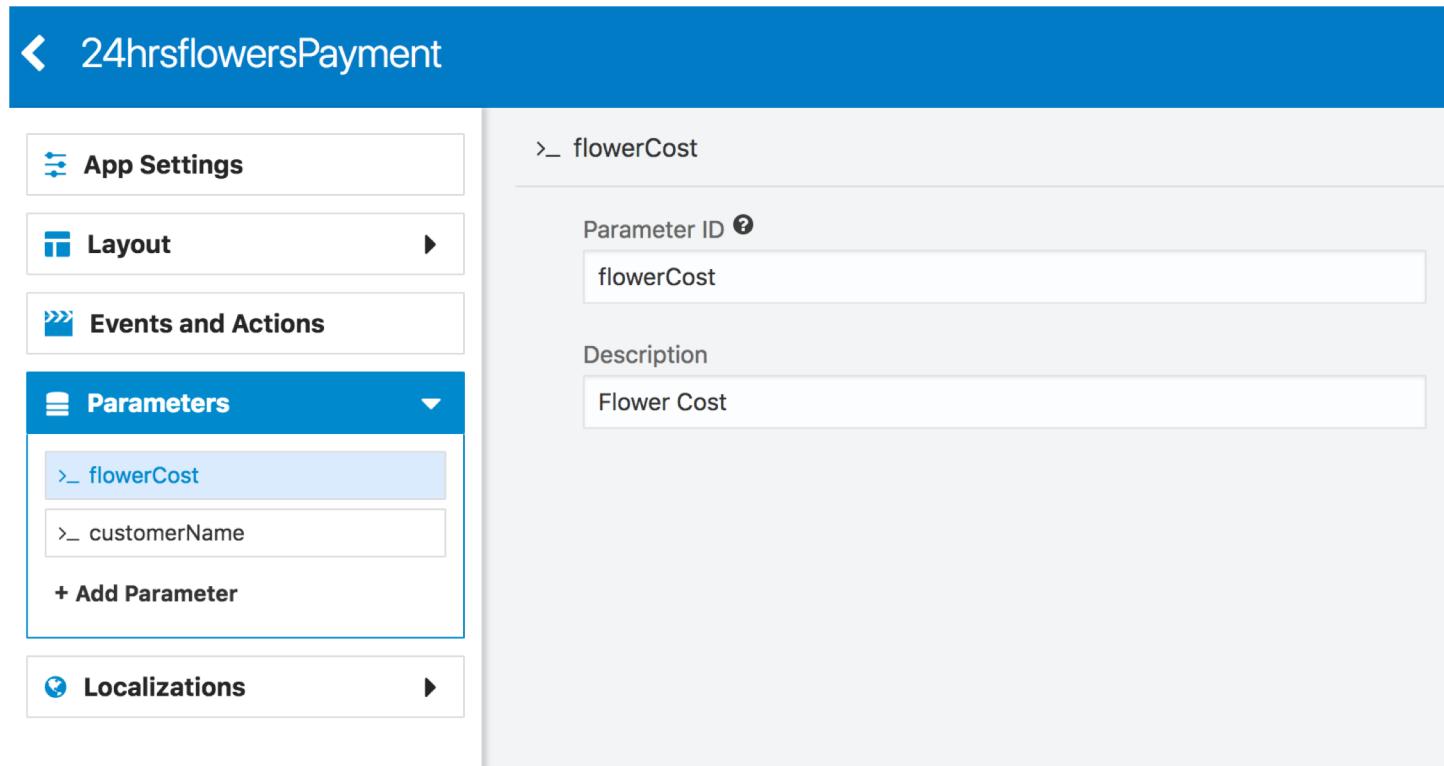
< 24hrsflowersPayment

- App Settings
- Layout
- Events and Actions
- Parameters
 - > flowerCost
 - > customerName
 - + Add Parameter
- Localizations

> flowerCost

Parameter ID ?
flowerCost

Description
Flower Cost



System.Interactive

editOrder:

component: "System.Interactive"

 properties:

id: "24hrsflowersEditOrder"

 prompt: "Please click the link below to ..."

 linkLabel: "Make Changes to the Card"

sourceVariableList: "senderName, senderMessage"

variable: "order"

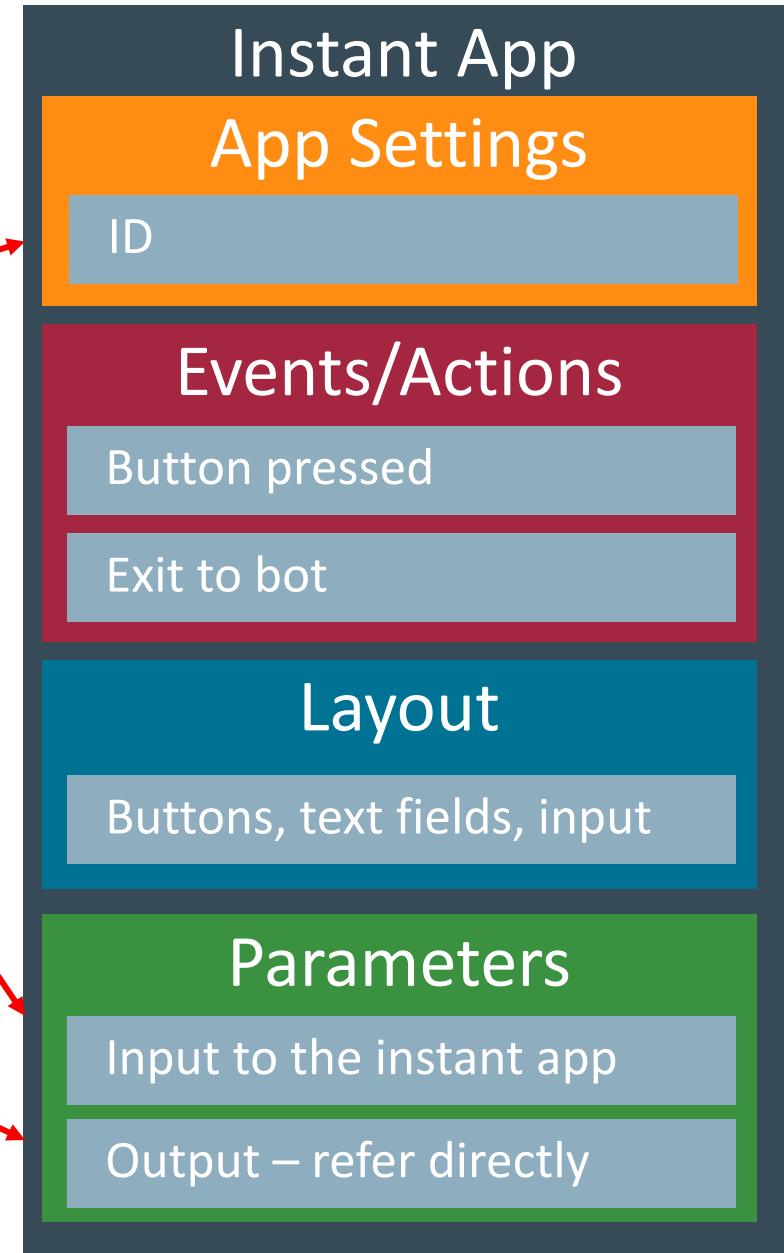
 cancelLabel: "Cancel"

transitions:

 actions:

 cancel: "orderCancel"

 textReceived: "handleTextInput"



System.Interactive

editOrder:

component: "System.Interactive"

 properties:

 id: "24hrsflowersEditOrder"

prompt: "Please click the link below to ..."

linkLabel: "Make Changes to the Card"

 sourceVariableList: "senderName, senderMessage"

 variable: "order"

cancelLabel: "Cancel"

transitions:

 actions:

cancel: "orderCancel"

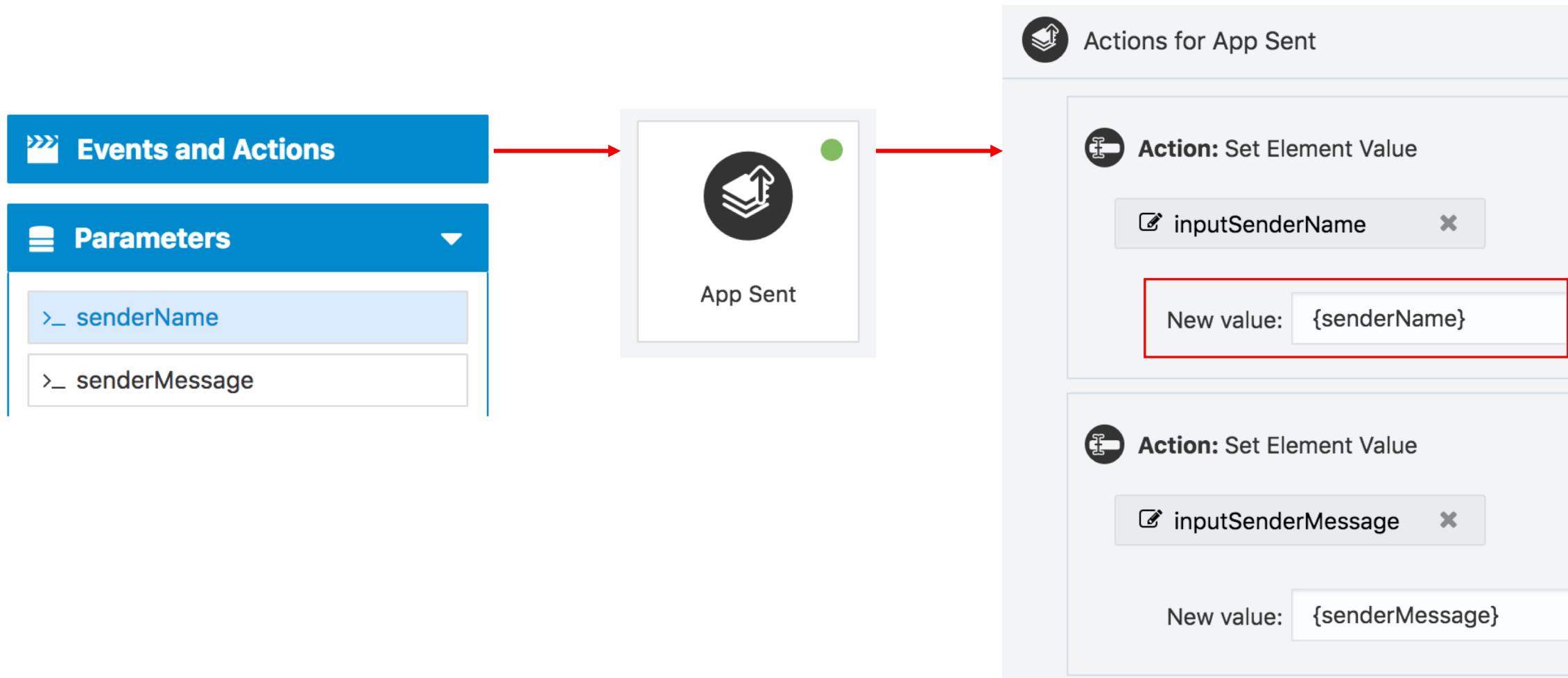
textReceived: "handleTextInput"

Please click the link below to make changes.

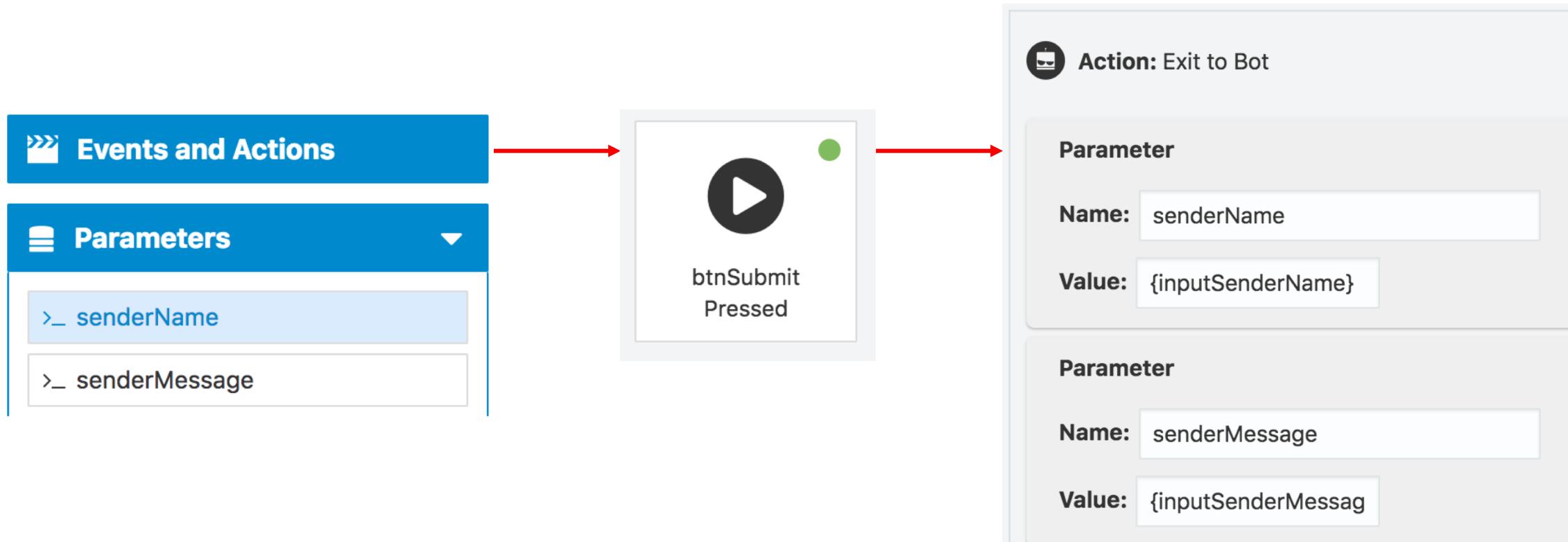
[Make Changes to the Card](#)

Cancel

Passing Parameters from Bot



Return Values from Instant App



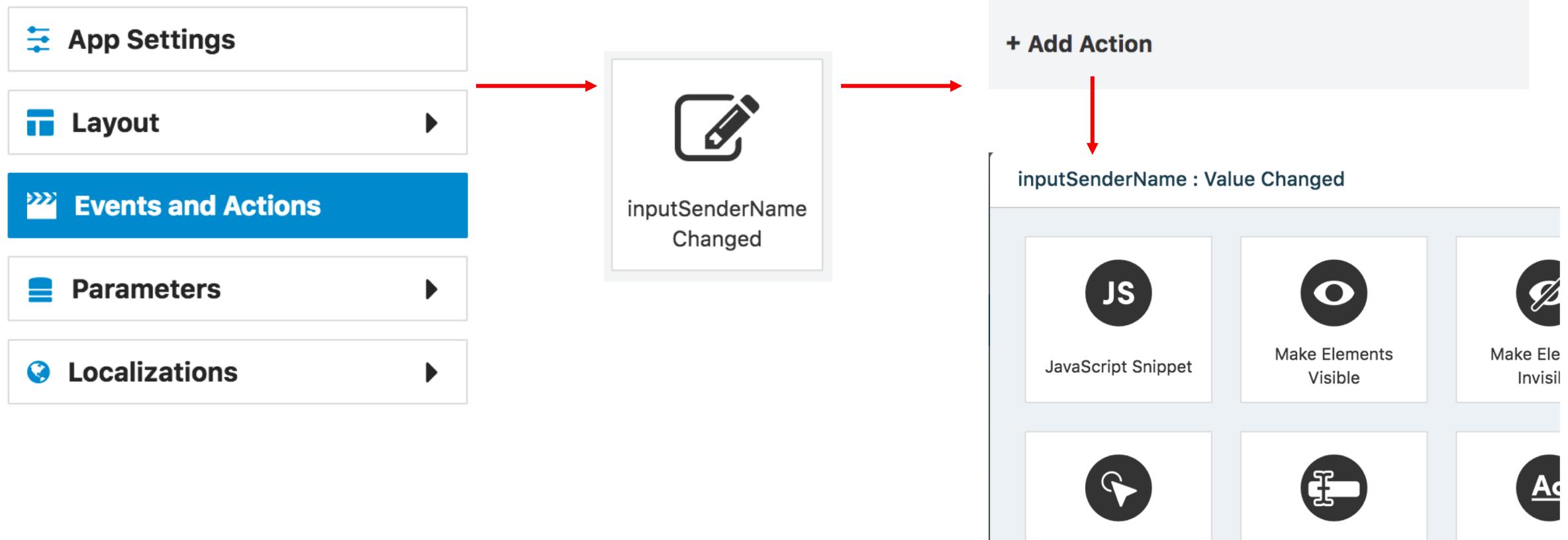
Accessing Return Values in the Dialog Flow

`${order.value.senderName}`

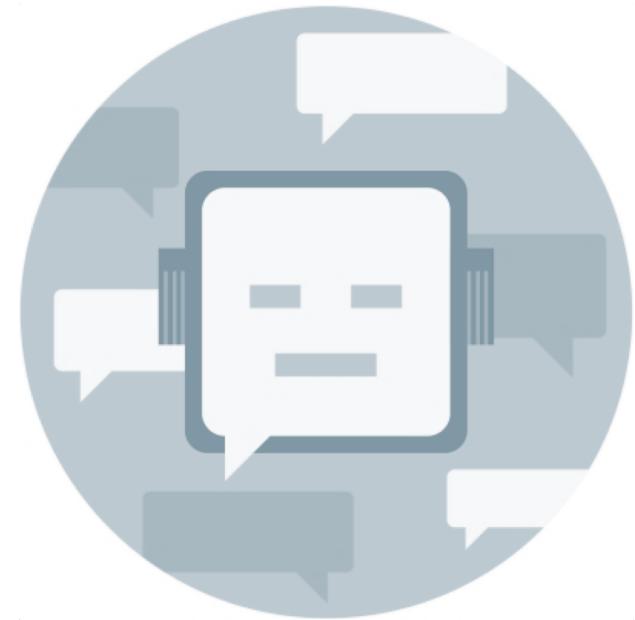
`${order.value.senderMessage}`

`${<variable>.value.<attribute>}`

Value Change Events



Instant Apps execute **on the server**



JavaScript Events

The screenshot shows a user interface for managing JavaScript snippets. On the left, a panel titled "JavaScript Snippet" contains a "JS" icon. A red arrow points from this icon to the "Action" panel on the right. The "Action" panel has a "JS" icon and the title "Action: Execute JavaScript Snippet". It includes a close button (X) and a status message "All Changes Saved" with a "Documentation" link. Below the title, a code snippet is displayed:

```
function valueChanged (app, chatbox, customer, element, oldValue,  
newValue) {
```

- **app** Access to the application context (parameters, elements)
- **chatbox** Handle to call external web sites, external service APIs, SMS app (on device)
- **element** UI component access (read / write attributes, show/hide etc.)
- **oldValue** Old component value
- **newValue** Changed, new component value

Example Function Calls

- `app.setElementValue(ElementId, value)`
 - `app.showElements(Elements)`
 - `app.hideElements(Elements)`
 - `app.enableElements(Elements)`
 - `app.disableElements(Elements)`
 - `app.toggleVisibility(Elements)`
-
- `app.setElementLabel(ElementId, textOrHTML)`
 - `app.resetElements(ElementName)`
 - `app.showAlertDialog(dialogTextOrHTML)`
 - `app.focus`
 - `app.exitToBot({key: value, key: value})`
`Element(ElementName)`

JavaScript Code Example

```
function callback (app, chatbox, statusCode, returnData, action) {  
    1 var result = JSON.parse(returnData);  
    2  
    3 var chkValid = result.valid;  
    4  
    5 if (chkValid)  
    6 {  
    7     app.hideElements("messageKO");  
    8     app.hideElements("Validate");  
    9     app.hideElements("reset");  
   10    app.showElements("button_submit");  
   11    app.showElements("messageOK");  
   12 }else if (!chkValid){  
   13     app.showElements("messageKO");  
   14     app.showAlertDialog("Your credit card number is not valid. ");  
   15 }  
   16 }  
}  
}
```



Advanced Bot Training Hands-On

Lab 7: Adding Instant Apps to your bot

Integrated Cloud Applications & Platform Services

ORACLE®