



Working Session

Build your own Orchestration



We will now be creating a simple approval orchestration,
from start to finish.

Phase 1

Preliminary Setup

Phase 2

Create
Orchestration

Phase 3

Celebrate!

Preliminary Setup

Get an environment

1. Get scratch org

Scratch Org

2. Please save email and login details

Salesforce DX Public Deployer

Your email address will be used in your new org. It's not stored anywhere else.

Email

Go

Salesforce DX Public Deployer

Note: It can take up to 5 minutes for the DNS to propagate. If the scratch org doesn't load properly, wait a few minutes to launch.

Your org is ready! Delete Launch

Your org will expire on **2022-08-14**.

Note: Use the Launch button above to login for the first time.
The credentials are for other uses (mobile app, integrations).
They auth to a sandbox (test.salesforce.com) not production (login.salesforce.com).
These credentials/button will continue working after the session-based Launch button stops.

Your username is **admin407@orchestrator.demo** and your password is **salesforce1** Launch via credentials

Preliminary Setup

Confirm your email address.

1. Check your email and confirm the change in email address to yours by clicking on the link.

Sandbox: Finish changing your Salesforce account's email address Inbox ×

 QA_SUPPORT@salesforce.com <qa_support@salesforce.com> Aug 17, 2022, 3:37 PM (20 hours ago) star
to me ▾

We recently received a request to change your Salesforce account's email address for username admin631@orchestrator.demo.

Previous email address: sfdx@mailinator.com
New email address: jbiserra@salesforce.com

To finish changing your email address, go to the following link. This link expires in 72 hours.
https://platform-saas-5840.my.salesforce.com/setup/emailverify?oid=00D2200000ERLI&k=Cj4KNQoPMDBEMjIwMDAwMDBFUkxsEg8wMkcyMjAwMDAwMDIRMXkaDzAwNTIyMDAwMDA0OG5pciAFGMih0u-qMBIQ_-odvcbER9F4R8luciTGeRoMoIElc4qUVTVCiDgUln0UXSazvfZ4N--zwFnfE8t7jGzlDovaKa7pcvqXAJiydwlyYZUoOAN3oe1baKGXovrdTuPwqtSwuttk9zljxYuPkcokwThnzt4ScVuyC4PlJlgKO887EVWrILJxb1ywgdGwS2IGvzbjEOFP3knSAP8I-6wRfHp7mAvmShuLw%3D%3D

If you can't click the link, copy and paste it into your Web browser. If you have questions, contact your Salesforce administrator.

Preliminary Setup

Configure Email

1. In your newly created org, click on the gear icon on the top right and select to **Setup**
2. Under Setup, on the far left search bar, type and select **Organization-Wide Addresses**
3. Create an Organization-Wide Email Address
 - a. Use your email address.
 - b. Select 'Allow All Profiles to Use this From Address'
4. Find the email in your inbox and open and click to confirm:



Edit Organization-Wide Email Addresses

An org-wide email address allows each user in a user profile to send email using this address. All users can also designate an org-wide email address for unmonitored mailboxes that require a verified address.

		<input type="button" value="Save"/> <input type="button" value="Save and New"/> <input type="button" value="Cancel"/>
Organization-Wide Email Address		
Display Name	Alex Address	
Email Address	einstein@salesforce.com	
Purpose	User Selection	
<input checked="" type="radio"/> Allow All Profiles to Use this From Address <input type="radio"/> Allow Only Selected Profiles to Use the From Address		

QA_SUPPORT@salesforce.com <qa_support@salesforce.com>
to me ▾ Wed, Aug 17, 3:39 PM (20 hours ago) ★


Dear User User,
We have received the following request to add this Salesforce Organization-Wide Email Address.
New Organization-Wide Email Address: jbiserra@salesforce.com

Click this link to confirm this Organization-Wide Email Address: https://platform-saas-5840.my.salesforce.com/setup/emailverify?oid=00D22000000ERLI&k=Cj4KNQoPMDBEMjlwMDAwMDBFUkxsEg8wMkcyMjAwMDAwMDIRMXkaDzAwNTIyMDAwMDA0OG5pcIAFGO2L1--qMBIQ78XwTSXp-8a1XQNQOyVALRoMW4GW0USRpkqH15Hqjm7aD6--ojQ6kyUm1otfondlOrzNdDQbeb4u_aJbNLAdkOoOdgrhObGckB8OC9mNig436JSxbpBhb2PowUL63jHGPbgpjjCimloGfz9HgssuX9BKyq9oRPMwLwuBiynLxqZixsD1Dada1nsTdDA%3D%3D

Troubleshooting Tip: If you don't see menu items in Setup, make sure you're not in Service Console, which has a simplified Setup menu

Preliminary Setup

Configure Email

Under Setup, on the far left search bar, type and select **Process Automation Settings** and set the ‘Automated Process User Email Address’ to the email address you’ve just verified:

Process Automation Settings

Help for this Page ?

Save Cancel

The default workflow user is required for scheduled paths in record-triggered flows and time-dependent actions in workflow rules. If the user who triggered the automation is no longer active, the default workflow user is associated with the actions that are executed by the automation. The default workflow user always runs schedule-triggered flows.

Default Workflow User

By default, no email address is set for the Automated Process User. Event processes and autolaunched flows with triggers use the email address to send emails for actions and errors. Orchestrations use the email address to send notifications. To set the email address for the Automated Process User, specify an organization-wide email address for the system administrator profile.

Automated Process User Email Address

You can now send email from your scratch org.

WARNING: Scratch orgs can only send 15 emails a day and then will fail silently.

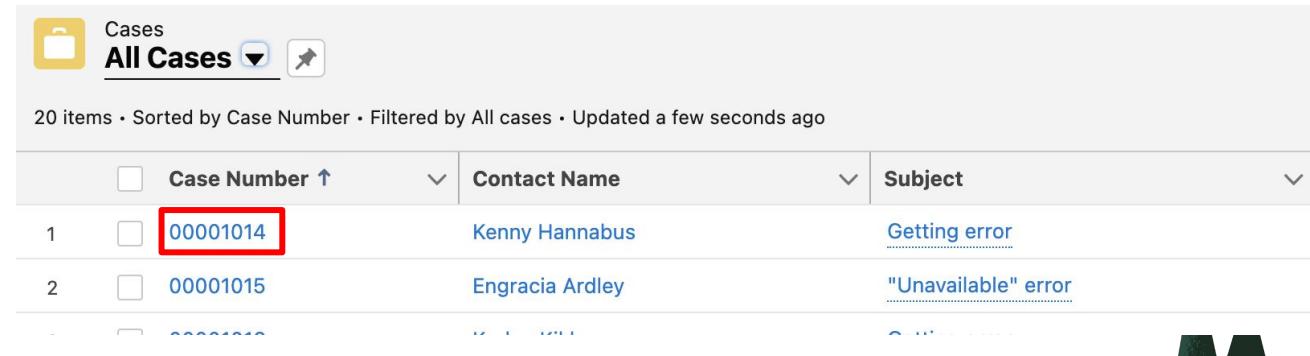
Preliminary Setup

Add the Work Guide component to your Case Page Layout

On the top left, click on the App Launch icon  and type and select **Cases**.

From the Cases listview dropdown, select **All Cases**

Open any Case by clicking on the Case number. **Make sure to keep this tab open! You will need it to test at the end!**

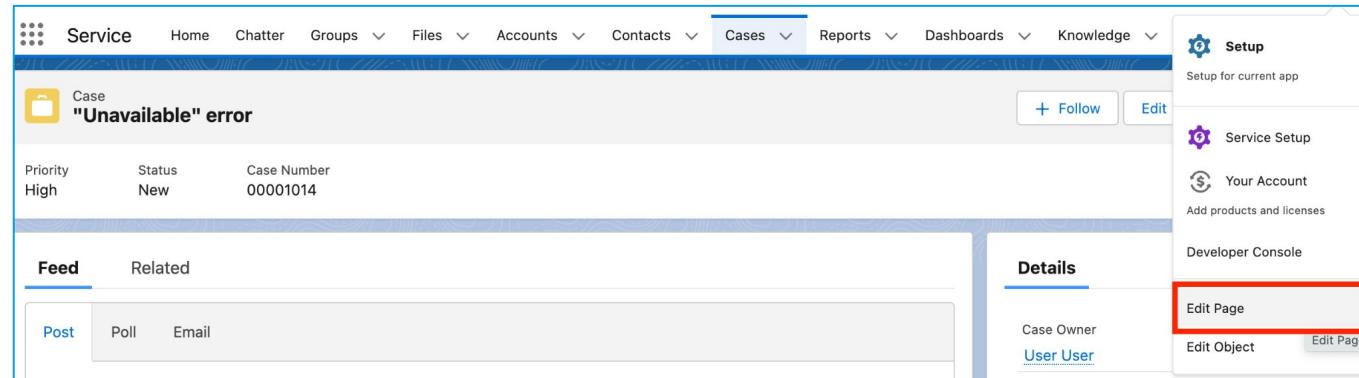


	<input type="checkbox"/> Case Number ↑	Contact Name	Subject
1	<input type="checkbox"/> 00001014	Kenny Hannabus	Getting error
2	<input type="checkbox"/> 00001015	Engracia Ardley	"Unavailable" error
-	<input type="checkbox"/> 00001016	Karen Kellie	Get it!

Preliminary Setup

Add the Work Guide component to your Case Page Layout

On the Case page click on the gear icon  on the top right and select “Edit Page”



In the Edit Page layout, type **Orchestrator Work Guide** and drag that component onto the right side of the screen above the Details component. Save and Activate the new page layout.



Preliminary Setup

Add the Work Guide component to your Case Page Layout

Make sure to assign your modified page to be the Org Default so Work Guide shows up everywhere. Set to Desktop only.

Once this has been set, click on the back arrow icon ← on the top left of the page layout editor.

Activation: Case Record Page

Custom record pages can be assigned at different levels:

- The org default** record page displays for an object unless more specific assignments are made.
- App default** page assignment, if specified, overrides the org default.
- App, record type, profile** assignments override org and app defaults.

[Learn more about Lightning page assignment.](#)

ORG DEFAULT **APP DEFAULT** **APP, RECORD TYPE, AND PROFI...**

Set this page as the org default to display it for all Case records, except when app default or app defined.

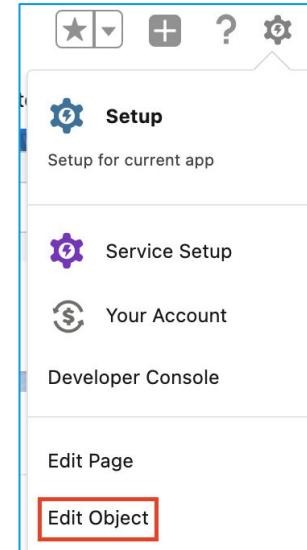
In standard Salesforce console apps, some objects have a system app default record page as the org default page, it doesn't display to users. To enable a custom org default page to show up as the app default. [Check your assignments.](#)

Assign as Org Default

Preliminary Setup

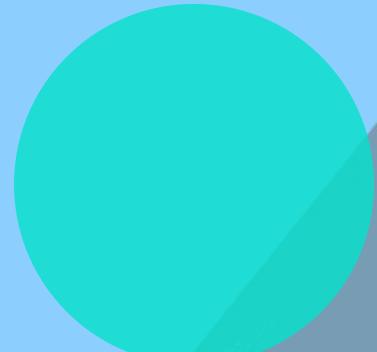
Add a ‘Refund Approved’ Case Status

- 1) Back on the Case layout, click on the gear icon and select “Edit Object”
- 2) On the object page select “Fields & Relationships”
- 3) Select the Status field
- 4) Click the New button in the Case Status Picklist Values section, and add a new ‘Refund Approved’ value.

A screenshot of the Case Field Status edit page. The page shows various configuration sections: Field Information, Picklist Values Used, Field Dependencies, Validation Rules, and Case Status Picklist Values. The 'Case Status Picklist Values' section is highlighted with a red box. It contains a table with four rows:

Action	Values	API Name	Closed
Edit Deactivate	New	New	<input type="checkbox"/>
Edit Del Deactivate	Working	Working	<input type="checkbox"/>
Edit Del Deactivate	Escalated	Escalated	<input type="checkbox"/>
Edit Del Deactivate	Closed	Closed	<input checked="" type="checkbox"/>

Let's Start Building



Building the Approval UI

To create a custom approval form for the approving manager, we'll start with a simple screen flow:

The screenshot shows a user interface for an approval request. At the top, it says "ApprovalScreenFlow". Below that is a section titled "Approval Request Response" with three radio button options: "Approved", "Approve with lower amount", and "Rejected". Underneath this is a label "Approved Amount" followed by an empty input field. At the bottom right is a blue "Next" button.

We'll connect this screen flow to an Interactive Step in our orchestration.

The key elements of the screen flow are:

- 1) the approval screen
- 2) passing the right information into the flow to show it on the screen (a case ID)
- 3) passing the approver's responses out of the flow to get it back to the orchestration

Building the Approval UI

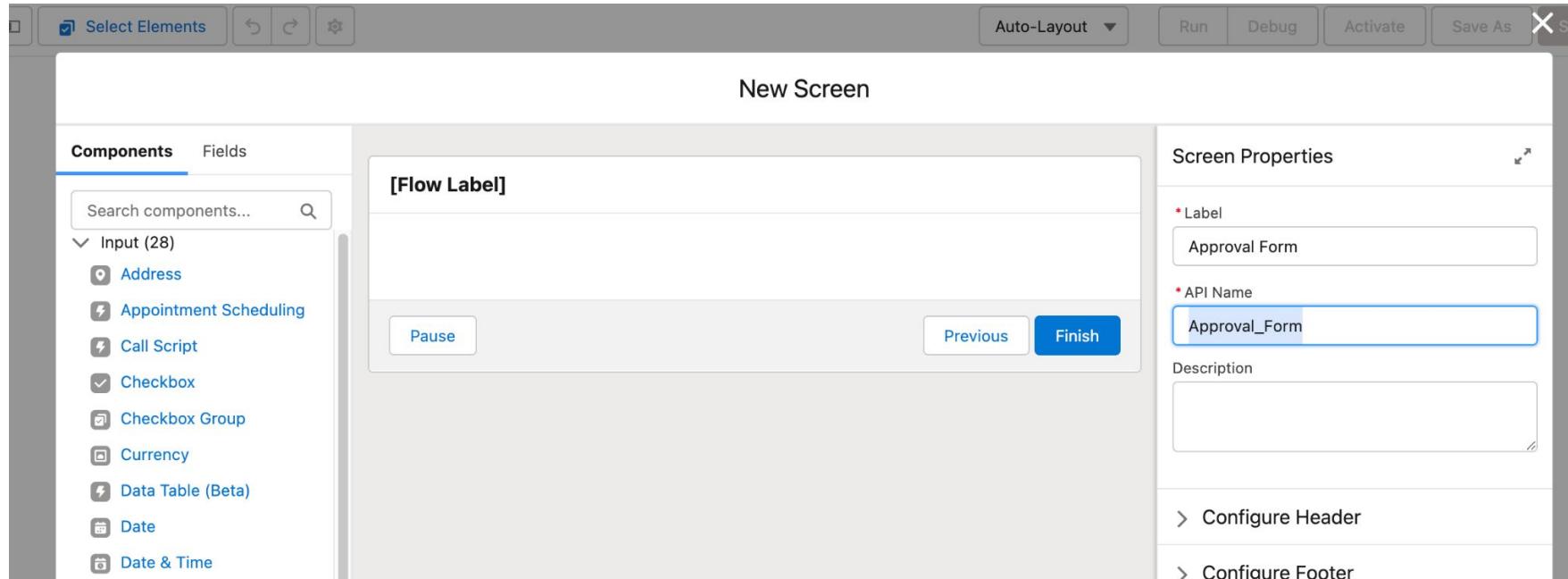
Step 1: On the top right, click on the gear icon and select **Setup**

Step 2: On the far left search bar, type and select **Flows**

Step 3: Press the “New Flow” button on the right side and select “**Screen Flow**” and click the “Create” button

Step 4: Build the Approval Form

- a) Click on the plus icon  between the Start and Stop elements of the flow to add a Screen element to the canvas and label it **Approval Form**

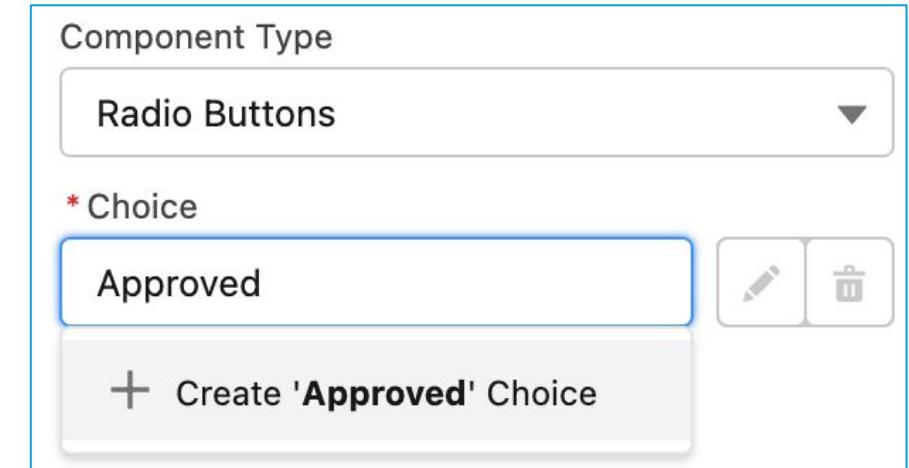
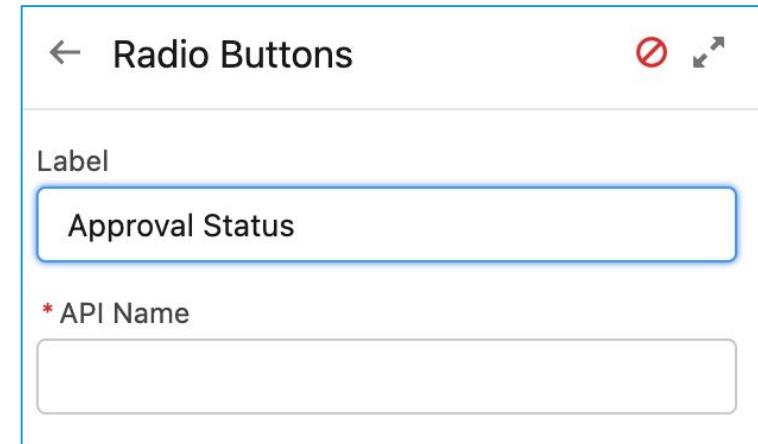


Building the Approval UI

On the far left, search for “Radio Buttons” under the components list. Drag the component to the screen canvas. Label the component “**Approval Status**”. The API name will automatically populate.

Create the following three Choices in the property editor on the right by typing the choice text in and then selecting ‘Create choice’ in the dropdown:

1. Approved
2. Rejected
3. Approved with lower Amount



Building the Approval UI

On the far left, search for “Currency” under the components list. Drag the component to the screen canvas. Label the component “Approved Amount”. The API name will automatically populate.

Click the **Done** button on the bottom right of the Edit Screen.

Edit Screen

[Flow Label]

* Approval Request Response

Approved

Approve with lower amount

Rejected

Number

Approved Amount

Decimal Places

← Number

Label

Approved Amount

* API Name

Approved_Amount

Require

Default Value

Enter value or search resources...

Decimal Places

Building the Approval UI

On the upper left click **Toggle Toolbox** to display the toolbox
Click on the “**New Resource**” button.

In the New Resource screen, populate with the following values:

- Resource Type: **Variable**
- API Name: **ApprovalStatus** (no spaces)
- Data Type: **Text**
- Select **Available for output**

Click the **Done** button on the bottom right of the screen

The requirement to explicitly specify which pieces of data in your flow can be accessed after it's done executing is a security safeguard.

The screenshot shows the Flow Builder interface. At the top, there is a header with a back arrow, a refresh icon, and the text "Flow Builder". Below the header, a toolbar has a "Select Elements" button with a red box around it. The main area is divided into two sections: "Toolbox" and "Manager". The "Manager" section contains a search bar with the placeholder "Search this flow..." and a "New Resource" button. Below this, a modal window titled "New Resource" is open, containing fields for "Resource Type" (set to "Variable"), "API Name" (set to "ApprovalStatus"), "Description" (empty), "Data Type" (set to "Text"), and "Default Value" (empty). At the bottom of the "New Resource" modal, there is a checkbox for "Allow multiple values (collection)" and a note about "Availability Outside the Flow" with checkboxes for "Available for input" and "Available for output", where "Available for output" is checked.

Building the Approval UI

On the far left of the flow builder, click on the “**New Resource**” button.

In the New Resource screen, populate with the following values:

- Resource Type: **Variable**
- API Name: **ApprovedAmount** (no spaces)
- Data Type: **Currency**
- Select **Available for output**

Click the **Done** button on the bottom right of the screen

The screenshot shows the Salesforce Flow Builder Manager interface. On the left, there's a sidebar with a search bar labeled "Search this flow..." and a "New Resource" button. The main area is titled "New Resource". It contains fields for "Resource Type" (set to "Variable"), "API Name" (set to "ApprovedAmount"), "Description" (empty), "Data Type" (set to "Currency" with a note about allowing multiple values for collections), "Decimal Places" (set to 0), "Default Value" (empty), and "Availability Outside the Flow" (with "Available for output" checked). At the bottom right of the dialog are "Cancel" and "Done" buttons.

Toolbox

Manager

Search this flow...

New Resource

New Resource

*Resource Type
Variable

*API Name
ApprovedAmount

Description

*Data Type
Currency

Allow multiple values (collection) i

Decimal Places
0

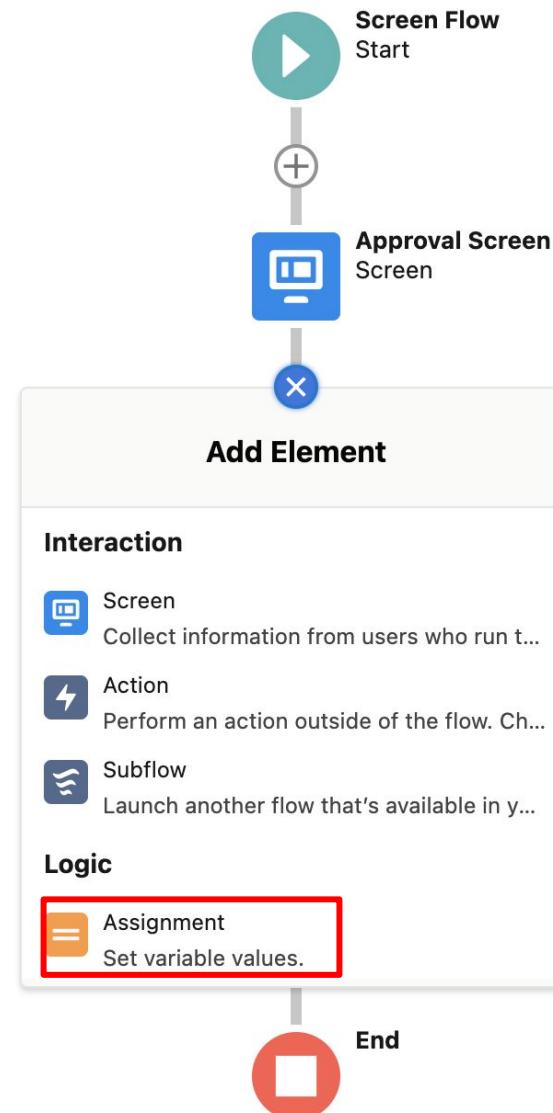
Default Value
Enter value or search resources...

Availability Outside the Flow
 Available for input
 Available for output

Cancel Done

Building the Approval UI

In the flow editor, add an **Assignment** element underneath your Approval Screen.



Building the Approval UI

In New Assignment screen, label it **Approval Outputs** (API Name will automatically populate)

Under the Set Variable Values, populate with the following:

- **ApprovalStatus EQUALS Approval_Status**
- **ApprovedAmount EQUALS Approved_Amount**

Click the **Done** button on the bottom right of the screen

New Assignment

* Label	Approval Outputs	* API Name	Approval_Outputs
Description			
Set Variable Values			
Each variable is modified by the operator and value combination.			
Variable	Operator	Value	
A_a ApprovalStatus X	Equals ▾	A_a Approval_Status X	Approval_Status X trash
Variable	Operator	Value	
A_a ApprovedAmount X	Equals ▾	Approved_Amount X	Approved_Status X trash
+ Add Assignment			

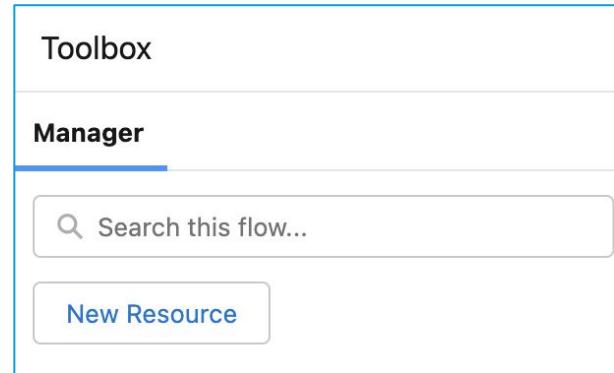
Building the Approval UI

On the far left of the flow builder, click on the “**New Resource**” button.

In the New Resource screen, populate with the following values:

- Resource Type: **Variable**
- API Name: **CaseID**
- Data Type: **Text**
- Select **Available for input**

Click the **Done** button on the bottom right of the screen

A screenshot of the "New Resource" configuration screen. The fields filled in are:

- * Resource Type: Variable
- * API Name: CaseID
- Description: (empty)
- * Data Type: Text
- Default Value: Enter value or search resources...
- Availability Outside the Flow:
 - Available for input
 - Available for output

The "Allow multiple values (collection)" checkbox is also present.

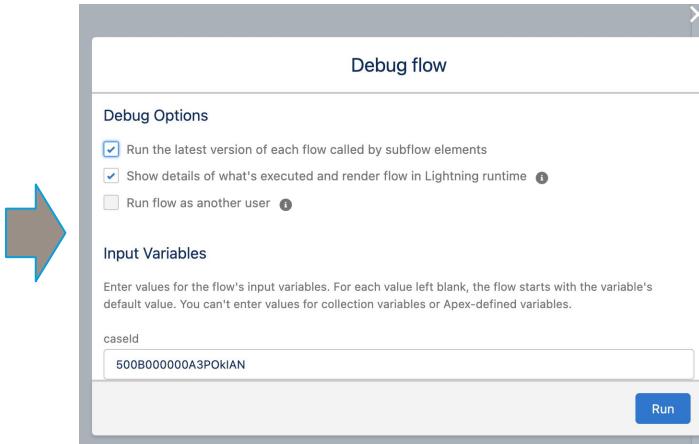
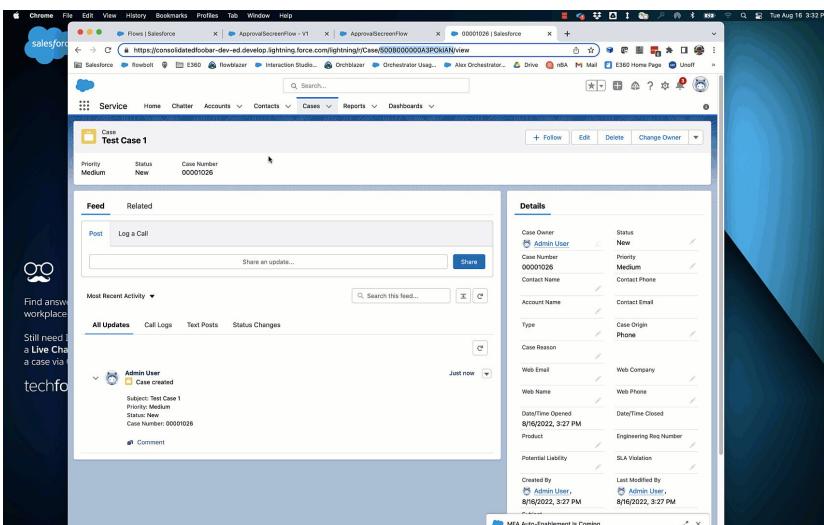
Building the Approval UI

In the flow editor, click on the Save button and label the flow as “**ApprovalScreenFlow**”.

Test it by clicking on the Debug button

a) Create a test case.

Copy the id from the case’s URL and paste it into ‘caseId’ in the Debug Flow window:



All done
The flow interview finished at Aug 16, 2022, 3:28:46 PM

Change Inputs Run Again

How the Interview Started
Admin User (005B0000008HszL) started the flow interview.
API Version for Running the Flow: 56
Some of this flow's variables were set when the interview started.
caseld = 500B000000A3POKIAN

Transaction Committed
Any records that the flow was ready to create, update, or delete were committed to the database.

SCREEN: Approval Form
Radio Buttons: Approval_Request_Response
Label: Approval Request Response
Data Type: Text
Choices selected at runtime:
Choice selected at runtime: Approve with lower amount (Approve_with_lower_amount)
Choice value: Approve with lower amount

Number: Approved_Amount
Label: Approved Amount
Value at run time: 400

Selected Navigation Button: NEXT

ASSIGNMENT: Prepare to Return to Orchestrator
={!approvedAmount} Equals {!Approved_Amount}
{!approverResponse} Equals
{!Approval_Request_Response}
Result
{!approvedAmount} = "400.00"
{!approverResponse} = "Approve with lower amount"

Goal: Your two output variables should reflect the values that you entered into the form:

Building the Approval UI

Click the Activate button to make the flow available to the orchestration.

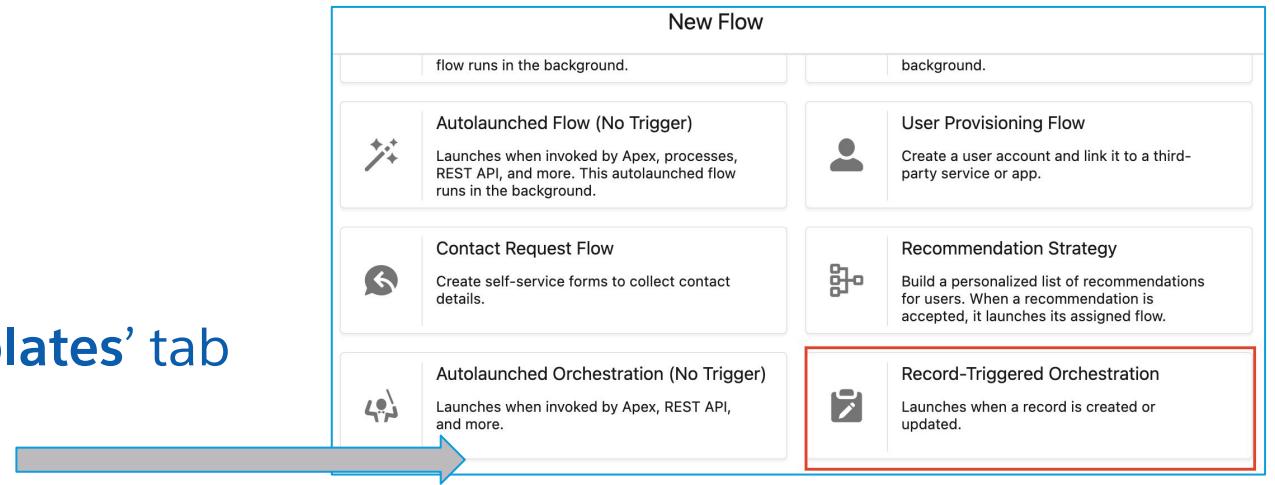
Take a Break!

You've completed the approval UI component of our automation!

Creating the Orchestration

Go to Setup → Flow

Click on New Flow and select the ‘All + Templates’ tab and select ‘Record Triggered Orchestration’



In the Start element and click **Edit**. Select ‘Case’ for your object



Click on the X icon on the top right corner of the Select Object screen.

Select Object
Select the object whose records trigger the orchestration when they're created or updated.
* Object
Case

Configure Trigger
* Trigger the Orchestration When:
 A record is created
 A record is updated
 A record is created or updated

Set Entry Conditions
Specify entry conditions to reduce the number of records that trigger the orchestration and the number of times the orchestration executions helps to conserve your org's resources.
Condition Requirements
None

Creating the Orchestration

On the flow canvas, add a new Stage and call it ‘Get Approval and Amount’

Click Save.

New Stage

* Label * API Name

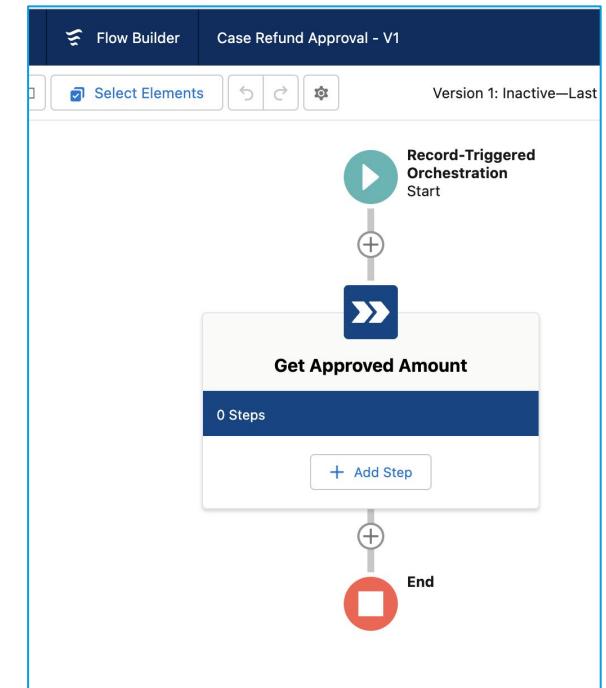
Get Approval and Amount Get_Approval_and_Amount

Description

Set Exit Condition

* Condition

When all steps have been marked Complete, the stage is marked Complete



Creating the Orchestration

Add a new Interactive Step to your Stage and label it '**Request Refund Approval**' with the following values:

- Flow: **ApprovalScreenFlow**
- CaseID: **{!\$Record.Id}** (this will automatically change to look like the screenshot)
- Record ID: **{!\$Record.Id}** (this will automatically change to look like the screenshot)
- Assignee Type: **User**
- User: **User User**

Notes: make sure to assign the step to your existing user account so you can easily find the notification email

Select a Flow

* Flow i

Approval Screen Flow

[Open Flow in Flow Builder](#)

Set Input Values

A_a CaseID

{!\$Record.Id}

Include

Specify Record Page

* Record ID i

A_a \$Record > Case ID

Specify Assigned User

* Assignee Type

User

* User

User User

Set Exit Condition

* Condition

When the assigned user has completed the screen flow, the step is marked Complete

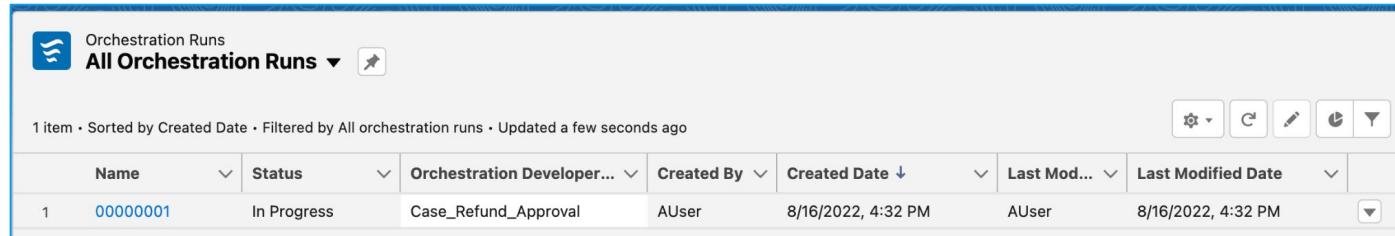
Test the Step

1. Click on the Save button on the top right and label your orchestration “**Case Refund Approval**”
2. **Activate** the orchestration
3. Double check that you followed all the steps above to enable email sending on your org
4. Open the tab with the original Case you opened and click on the **Clone** button.
5. Check your inbox! You should have a notification .
 - a. NOTE: you’re currently playing the role of the Approver
6. Click on the notification. You should be taken to the Case page

Tracking & Troubleshooting

Whether you're tracking down a problem or just want to see what's going on, the Orchestration list views provide a top-down view

Step 1: Check the Orchestrator Runs list view (Click the App Launcher, type **Orchestration Runs**) to see the Orchestration Run that has been created:

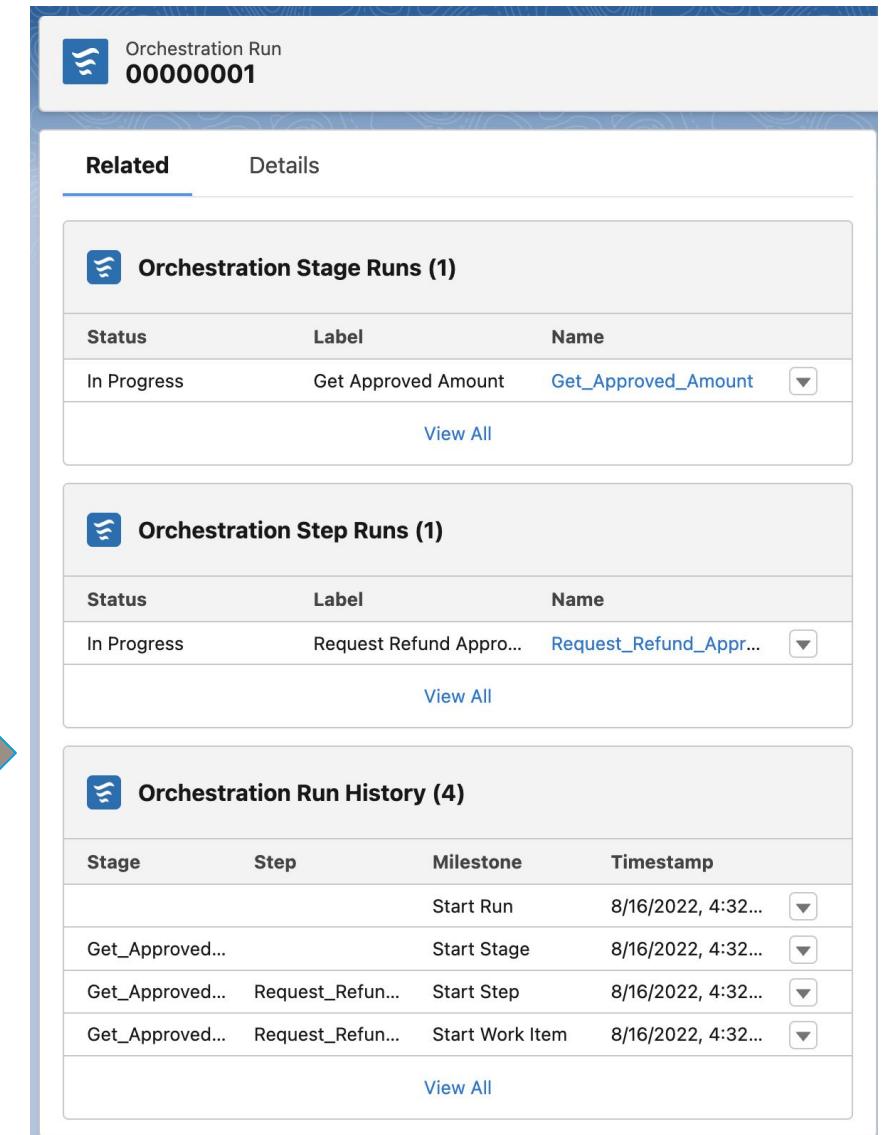


The screenshot shows a list view titled "All Orchestration Runs". It displays one item with the following details:

Name	Status	Orchestration Developer...	Created By	Created Date	Last Mod...	Last Modified Date
00000001	In Progress	Case_Refund_Approval	AUser	8/16/2022, 4:32 PM	AUser	8/16/2022, 4:32 PM



Step 2: Visit your Orchestration Run, click on Related, and explore the History, Stage, and Step information:



The screenshot shows the details page for an Orchestration Run. At the top, it displays the Run ID: 00000001. Below that, there are three tabs: "Related", "Details", and "History". The "Related" tab is selected, showing three sections: "Orchestration Stage Runs (1)", "Orchestration Step Runs (1)", and "Orchestration Run History (4)".

Orchestration Stage Runs (1)

Status	Label	Name
In Progress	Get Approved Amount	Get_Approved_Amount

Orchestration Step Runs (1)

Status	Label	Name
In Progress	Request Refund Appro...	Request_Refund_Appr...

Orchestration Run History (4)

Stage	Step	Milestone	Timestamp
Get_Approved...		Start Run	8/16/2022, 4:32...
Get_Approved...	Request_Refun...	Start Stage	8/16/2022, 4:32...
Get_Approved...	Request_Refun...	Start Step	8/16/2022, 4:32...
Get_Approved...	Request_Refun...	Start Work Item	8/16/2022, 4:32...

Tracking & Troubleshooting

Step 3: Drill down into the ‘In Progress’ Step Run to find the Work Item and see who it is assigned to:

Orchestration Step Run
Request_Refund_Approval

Related Details

Orchestration Work Items (1)

Status	Step	Name	Description
Assigned	Request Refund...	00000001	

[View All](#)



Orchestration Work Item
00000001

Related Details

Information

Status
Assigned

Orchestration Step Run
[Request_Refund_Approval](#)

Assigned To
 [User2](#) [User2](#)

Screen Flow
300B00000009KiYIAU

Description

Step
Request Refund Approval

Context Record
[00001027](#)

Last Modified By
 [Automated Process](#), 8/16/2022, 4:32 PM

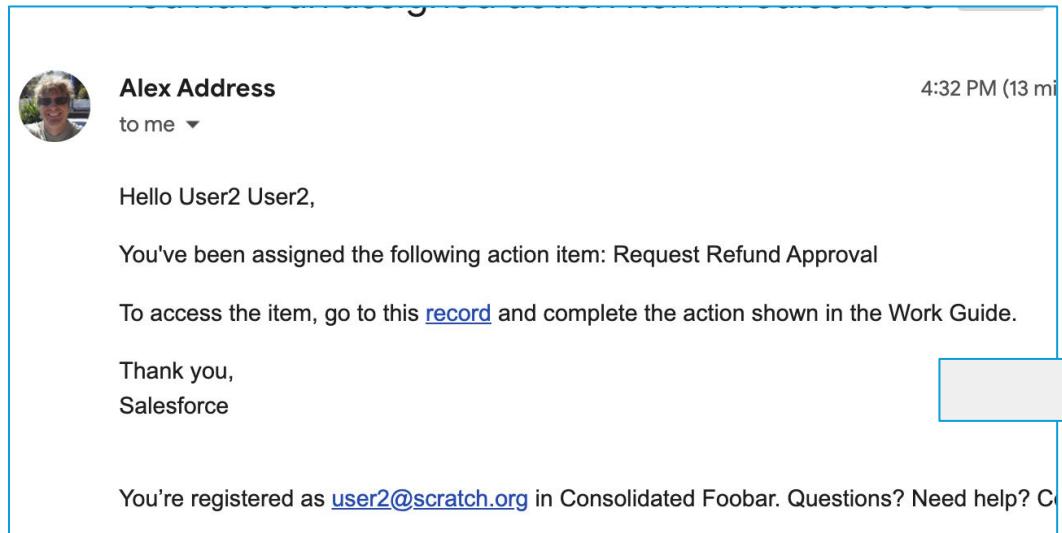
Created By
 [Automated Process](#), 8/16/2022, 4:32 PM

Created Date
8/16/2022, 4:32 PM

Looking for Your Work Items

If you're not seeing work show up in the Work Guide, check to confirm that the User that you're logged in as is the same as the User that has the work assigned to them. They might be different Users that use the same email.

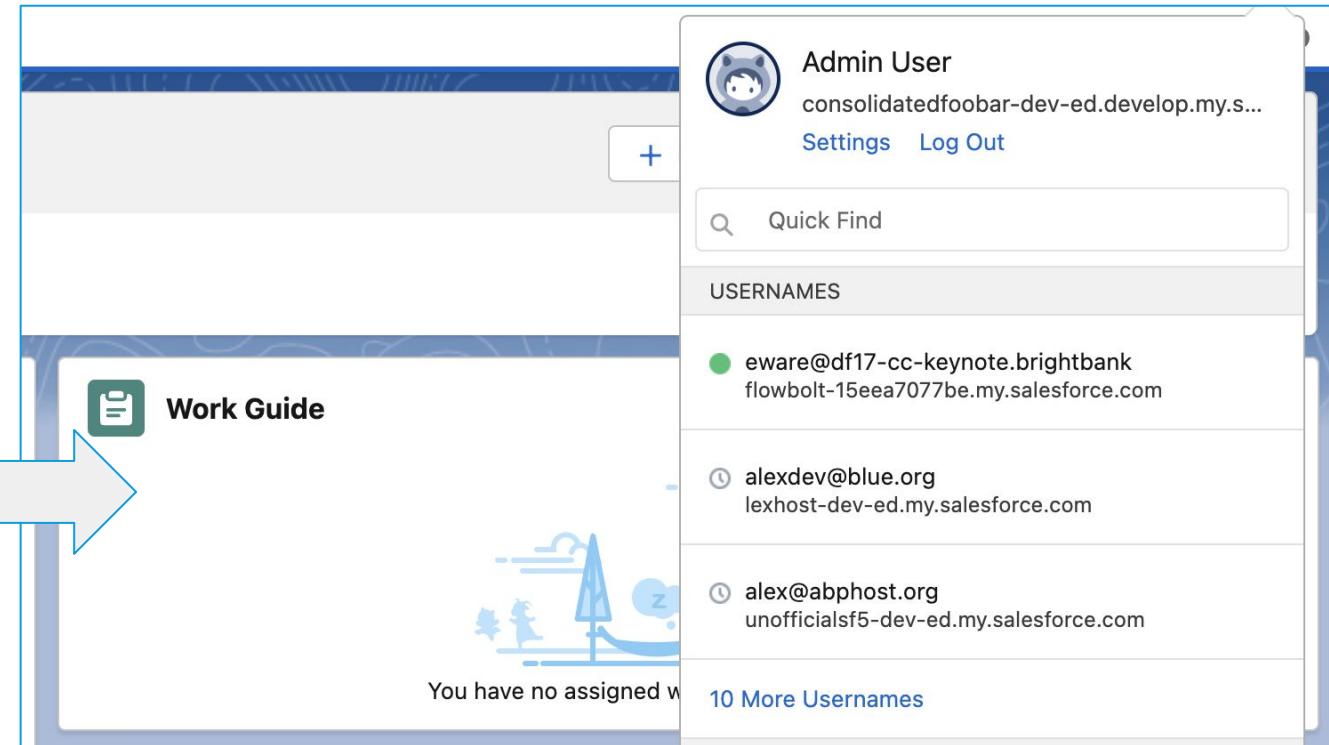
Here, the work is assigned to User2 User but I'm logged in as Admin User, so I don't see any work.



A screenshot of an email inbox. A single email message is selected, highlighted with a blue border. The message is from "Alex Address" to "me". The subject line is "Hello User2 User2,". The body of the email contains the following text:

Hello User2 User2,
You've been assigned the following action item: Request Refund Approval
To access the item, go to this [record](#) and complete the action shown in the Work Guide.
Thank you,
Salesforce

You're registered as user2@scratch.org in Consolidated Foobar. Questions? Need help? C



A screenshot of the Salesforce Work Guide interface. At the top, there's a navigation bar with the user "Admin User" (consolidatedfoobar-dev-ed.develop.my.s...), "Settings", and "Log Out". Below the navigation is a search bar labeled "Quick Find". The main area is titled "Work Guide" with a green icon. It displays the message "You have no assigned w..." followed by "10 More Usernames". On the left side of the main area, there's a large blue arrow pointing from the "Work Guide" section towards the left edge of the screen, indicating a connection to the email message above.

Visiting the Work Item

Not all orgs have Login As Different Users enabled. You can turn this on in Setup → Login Access Policies

SETUP

Login Access Policies

Control which support organizations your users can grant login access to.

Manage Support Options

Setting	Action	Enabled
Administrators Can Log in as Any User	<input checked="" type="checkbox"/>	



Action	Full Name ↑	Alias	U
<input type="button" value="Edit"/>	Chatter Expert	Chatter	C
<input type="button" value="Edit"/>	User, Admin	AUser	V
<input type="button" value="Edit"/>	User2, User2	uuser	U

Being able to quickly switch between Users is a huge aid to building multi-user orchestrations

Visiting the Work Item

Once you confirm that the Interactive Step is properly showing up in the Work Guide, it's time to return to the Orchestration and do something useful with the Approver's responses

The screenshot shows a 'Work Guide' interface for a 'Case Refund Approval > Get Approved Amount' step. At the top, there are buttons for '+ Follow', 'Edit', 'Delete', 'Change Owner', and a dropdown menu. On the left, there is a vertical sidebar with a 'Share' button and other icons. The main area is titled 'Request Refund Approval'. It contains a section for 'Approval Request Response' with three radio buttons: 'Approved', 'Approve with lower amount', and 'Rejected'. Below that is a field labeled 'Approved Amount' with an input field. At the bottom right is a 'Next' button.

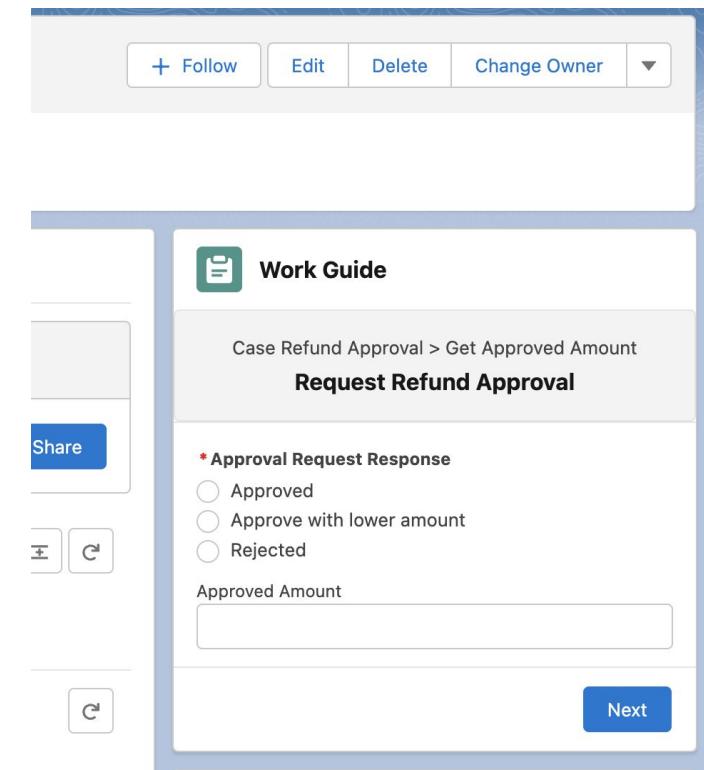
Take a Break!

You're successfully dispatching targeted work to assignees.

Handling the Approver's Responses

In our example, we'll implement the following business rule:

- 1) if the request is Approved unconditionally or Approved to a specific limit, change the Case Status to “Refund Approved” (This would presumably trigger additional automation, either in the form of additional Steps in this orchestration, or in some other part of Salesforce.)



Implementation of the handling logic for the Rejection path is outside the scope of this workshop, but feel free to experiment with ideas there, like sending an email back to the person who created the case.

Updating the Case Status

- 1) To update the Case Status, we need to create a new Background (Autolaunched) Flow that will execute an Update Record.

a) This flow is NOT a triggered flow so we create it without a trigger. It will get invoked by the orchestration.



The screenshot shows the 'New Flow' screen in the Flow Manager. Under the 'Core' tab, there are four options: 'Screen Flow', 'Record-Triggered Flow', 'Schedule-Triggered Flow', and 'Autolaunched Flow (No Trigger)'. The 'Autolaunched Flow (No Trigger)' option is highlighted with a blue box and a blue arrow pointing to it from the text above. A 'Create' button is visible at the bottom right.

b) As in our first flow, we need to create a **text variable**, labeled **caseId** and mark it '**Available for Input**' so the orchestration can pass in the Case Id. We'll call this one 'caseId' like we did before.

The screenshot shows the 'New Resource' dialog for creating a variable. The 'Resource Type' is set to 'Variable', 'API Name' is 'caseld', 'Data Type' is 'Text', and 'Default Value' is empty. The 'Availability Outside the Flow' section has the 'Available for input' checkbox checked. At the bottom are 'Cancel' and 'Done' buttons.

Updating the Case Status

- 1) Add an Update Records element and have it use the caseId that the Orchestrator will provide to update the Status value

Label: Change Case Status to "Refund Approved"

Object: Case

All Conditions Are Met (AND)

Field: Id

Operator: Equals

Value: caseId

Set Field Value for the Case Records

Field: Status

Value: enter Refund Approved

New Update Records

Change Case Status to 'Refund Approved' Change_Case_Status_to_Refund_Approved

Description

* How to Find Records to Update and Set Their Values
 Use the IDs and all field values from a record or record collection
 Specify conditions to identify records, and set fields individually

Update Records of This Object Type
* Object
Case

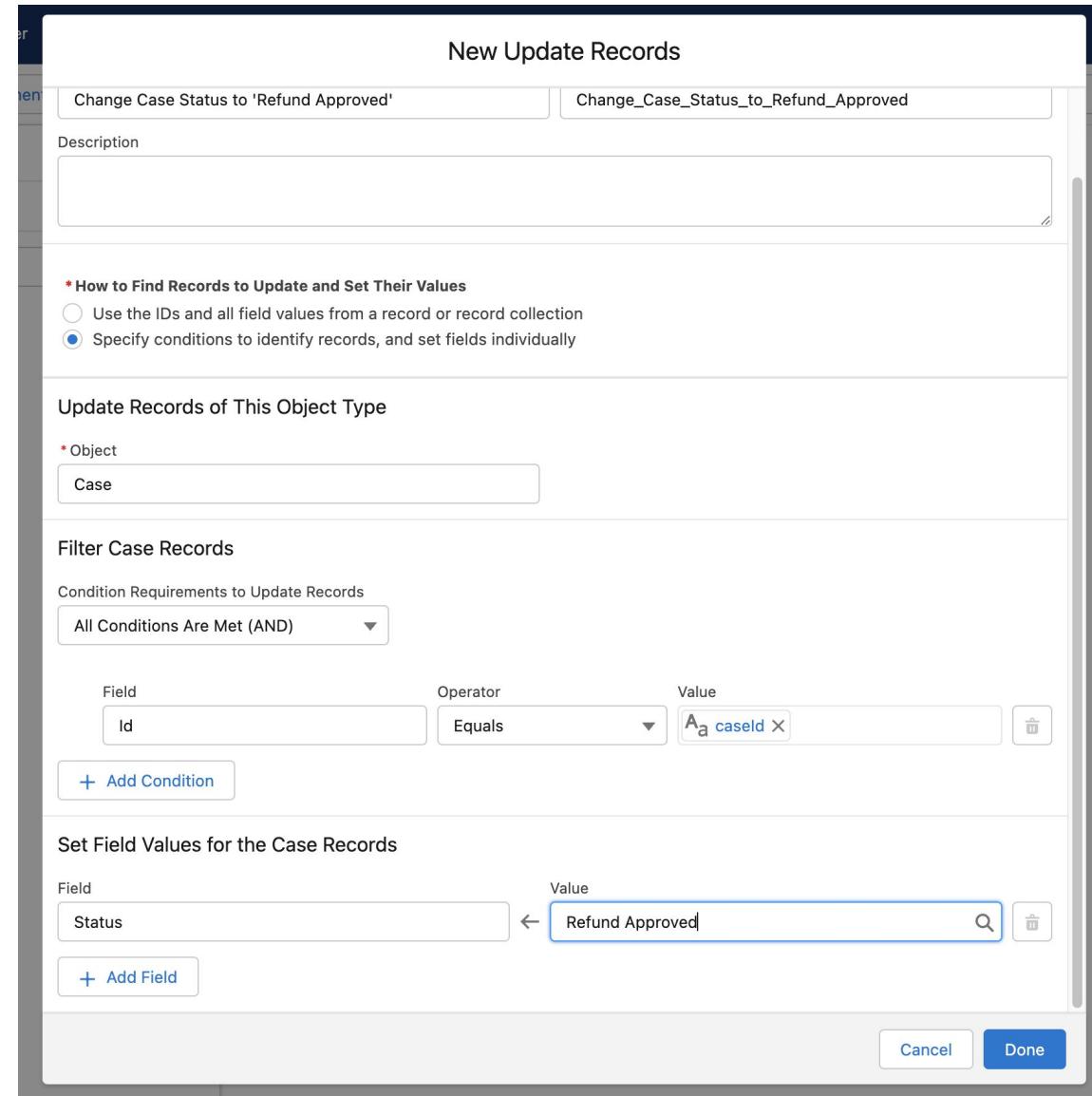
Filter Case Records
Condition Requirements to Update Records
All Conditions Are Met (AND)
Field Operator Value
Id Equals Aa caseld X

+ Add Condition

Set Field Values for the Case Records
Field Value
Status Refund Approved

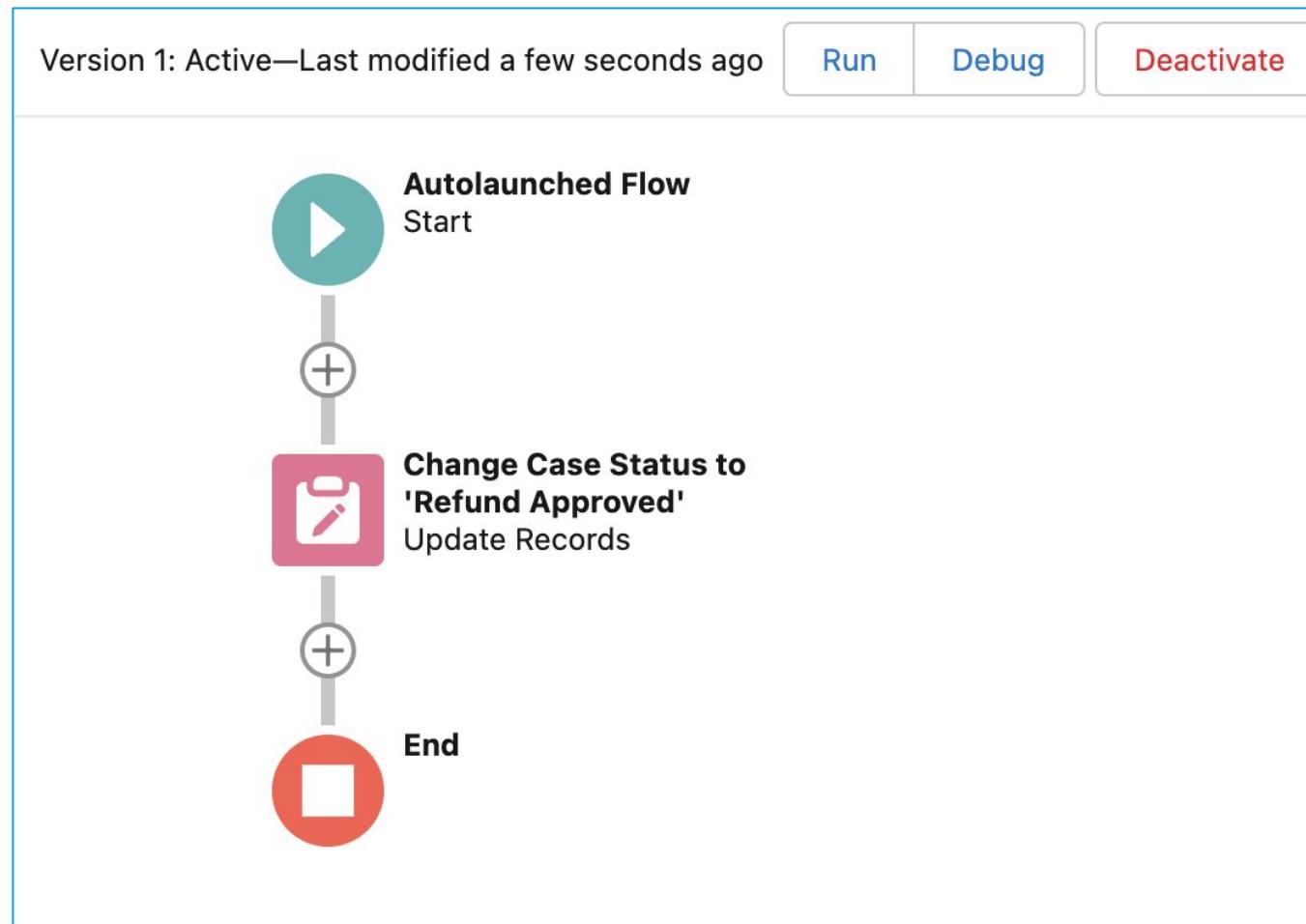
+ Add Field

Cancel Done



Updating the Case Status

1. Save your flow as **UpdateCaseStatus** and Activate it



Create a Branch in Your Orchestration

In your Orchestration, add a **Decision** element under **Get Approval and Amount** that directs the flow to one of two new stages. One path should be taken if the orchestration was approved, and the other should be taken if it was rejected.

Decision Label: **Approved**

Outcome Details:

Label: Change New Outcome to Yes

Condition Requirements to **Any Condition is Met (OR)**

Resource: `{!Request_Refund_Approval.Outputs.ApprovalStatus}`

Operator: **Equals**

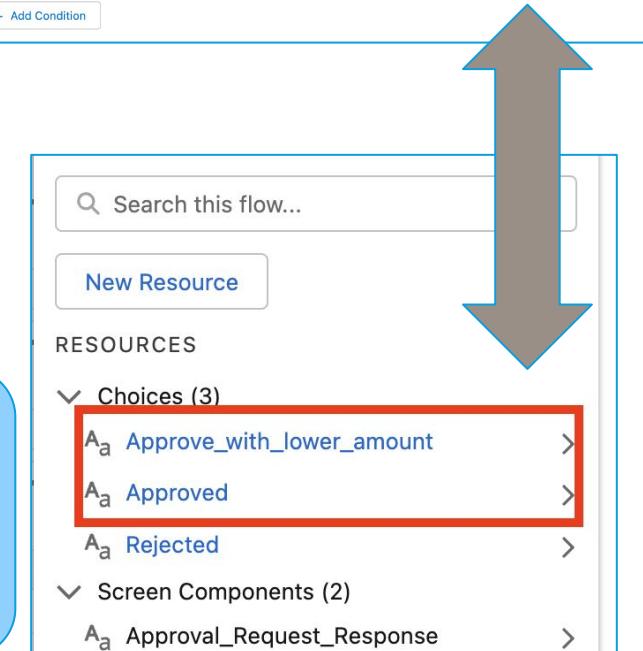
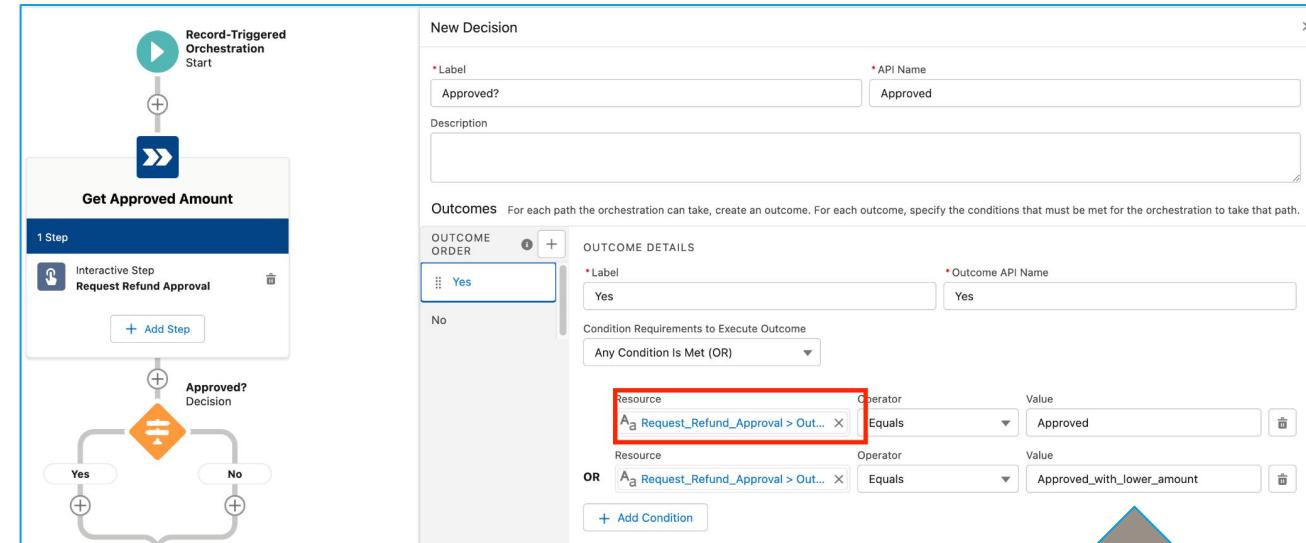
Value: enter **Approved**

Change "Default Outcome"

Label = **No**

The Decision element examines the Outputs of the Request Refund Approval Step

The values it's examining are the original choices that were defined in the interactive step



Add New Stages

Add stages called ‘Process Refund’
and
‘Deny Refund or Resubmit’



Add the Update Case Status Step

We only want to update the Case Status to ‘Refund Approved’ if the orchestration routes into the ‘Process Refund’ stage, so we’ll add a Background Step there that runs as soon as that stage starts.

Here again (and for almost every Step), we want to pass in the id of the main ‘context’ record.

Add a **Background Step** to Process Refund

Label: **Update the Case**

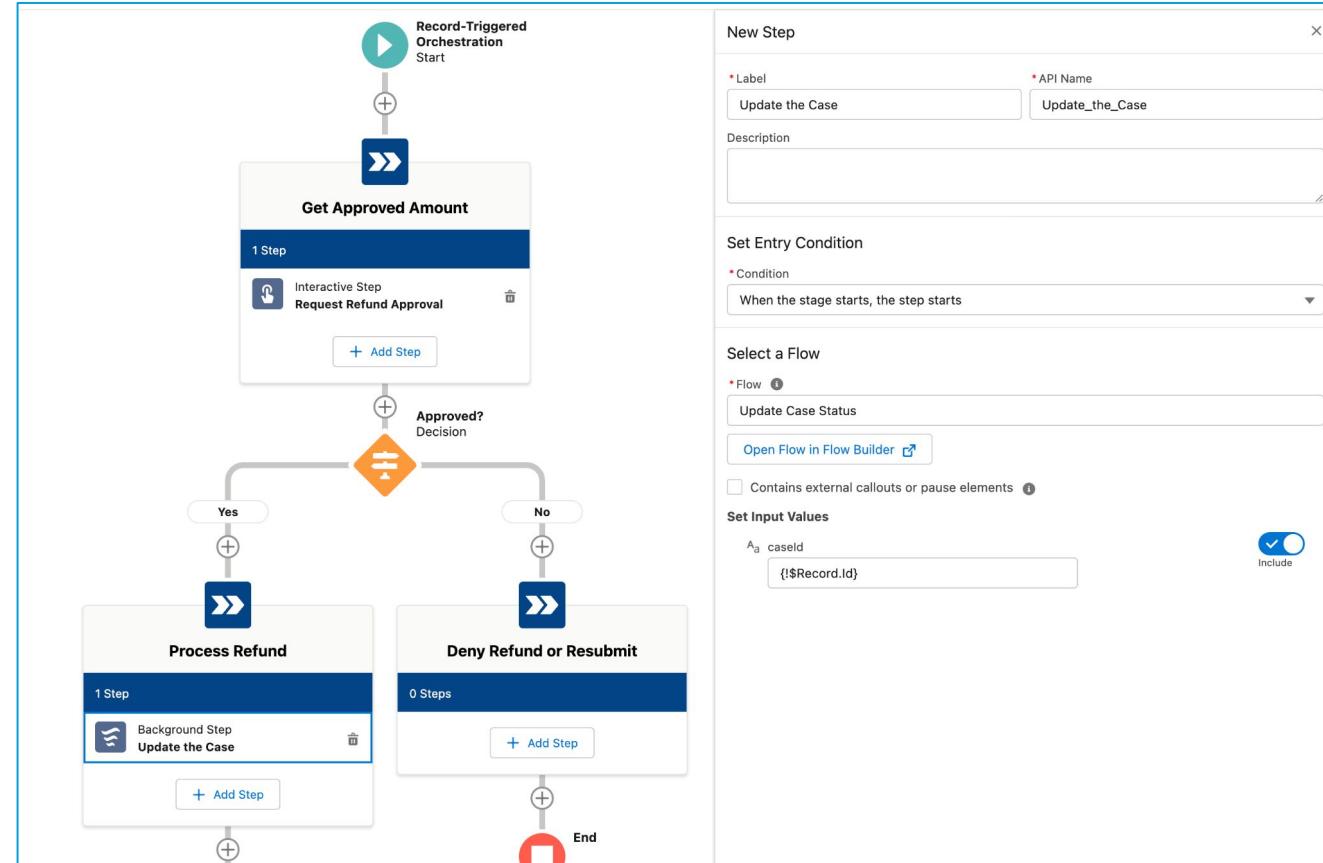
Flow: **Update Case Status**

*** If the flow doesn’t show in the list, save and refresh the builder page

Click the **Include** toggle

Input Value = `{!Record.Id}`

Save & Activate



Final Testing

1. Create a case
2. Verify that the 'Request Refund Approval' assignee receives an email notification
3. Logging in as that assignee, fill out the Approval form and approve the refund request.
4. Verify that the Case Status is then updated to 'Refund Approved'
5. Create another case, but this time, reject the refund request

THANK YOU

