# 2. Takeoff Checklist 🚀

Welcome to the next phase of our workshop! This session, spanning over two days, builds upon the work you've done previously in creating the Telework Case Management application. Some of you might have developed this application during yesterday's session, while others may have created it in a prior workshop. Regardless of when you built it, we'll start by ensuring everyone has the same foundational application.

### Let's make sure you have everything you need to get started.

Before beginning the lab exercies, you should have the following:

- [ ] A ServiceNow instance URL and credentials.
- [ ] Download the lab files bundle: Telework_Lab_Files.zip (Right-click and select 'Save link as...')
- [ ] Extract the `Telework_Lab_Files.zip` file:
    - MacOS users:
        - Right-click on the `Telework_Lab_Files.zip` file.
        - Choose “Extract All” or “Open with Archive Utility” (the exact option may vary depending on your macOS version).
        - The files will be extracted into a folder named `Telework_Lab_Files`.
    - Windows users:
        - Right-click on the `Telework_Lab_Files.zip` file.
        - Select “Extract All”.
        - In the dialog that appears, choose where you want the files to be extracted or use the default location.
        - Click “Extract” to unpack the files into the specified folder.

After extraction, the `Telework_Lab_Files` folder should contain the following files:

- Telework_Case_Management_v1.0.5.xml
- people finder.png
- Portal_Background.png
- SHIELD_Logo.png

# Import the Telework Application

In this exercise, we will learn how to import an existing scoped application using an Update Set, or xml file. This application was created in the 101 lab, Telework Case Management.

## Steps to Importing Application

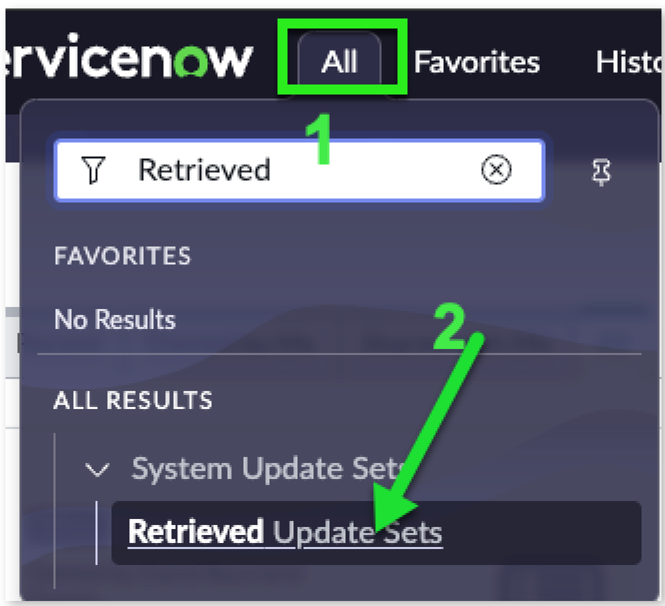
1. Go to 'Retrieved Update Sets'
2. Scroll to 'Releated links' and click on 'Import Update Set from XML'
3. Upload Update Set
4. Find the record you just imported: State = 'Loaded'
5. Preview Update Set
6. Accept/Skip Remote Updates
7. Commit Update Set

Here are those steps in detail:

## Step 1 - Go to 'Retrieved Update Sets'

1. Go to All

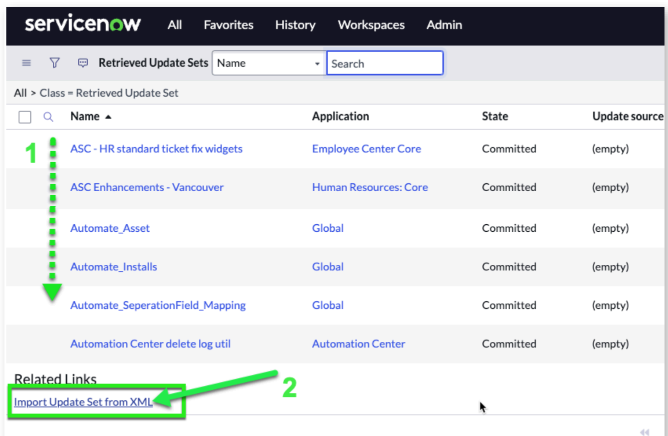
2. Then under 'System Update Sets' go to: Retrieved Update Sets



# Step 2 - Import Update Set

1. Scroll to the bottom of the list

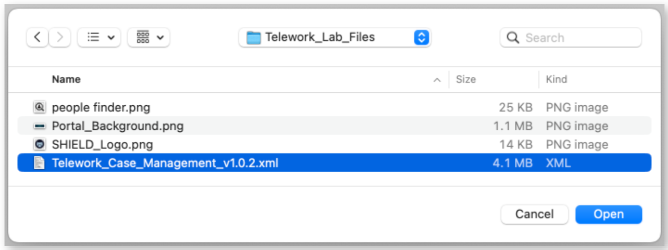
2. Click on related link: Import Update Set from XML




# Step 3 - Upload Update Set

1. Click on 'Choose File'

2. Find and choose the file you downloaded earlier "Telework Case Management...xml"

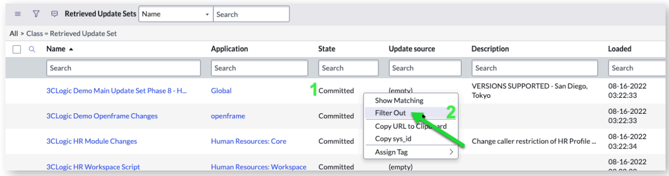



3. Click on 'Upload'

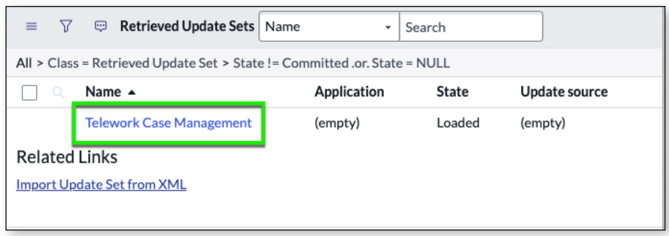

# Step 4 - Find record State = 'Loaded'

1. Right click on 'Committed' (any record)

2. Choose "Filter Out"

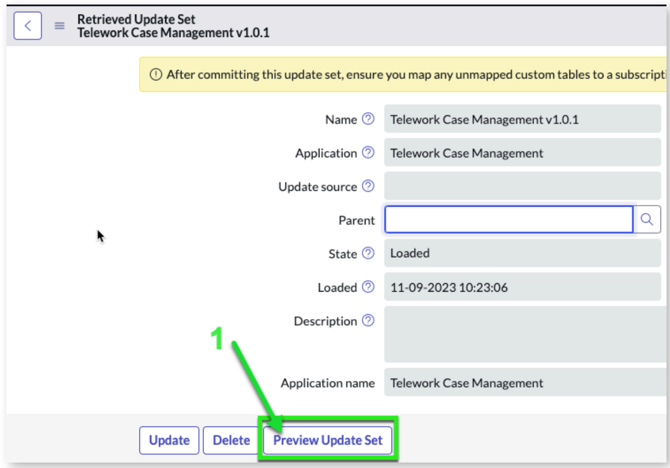



3. Click on retrieved update set "Telework Case Management..."



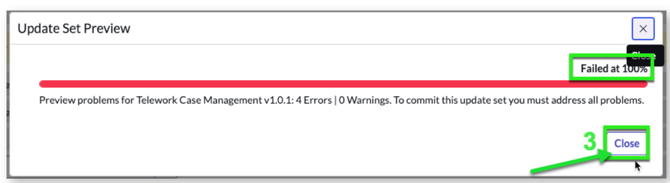

# Step 5 - Preview Update Set

1. Click on 'Preview Update Set'




2. Watch the bar progress...until 100%

3. Click on 'Close' (there can be errors, so it's ok if it says 'Failed')




## Step 6: if there are errors. If no errors, go to Step 7

For more information regarding update sets and use of the different options, here are some helpful links:

ServiceNow Update Sets

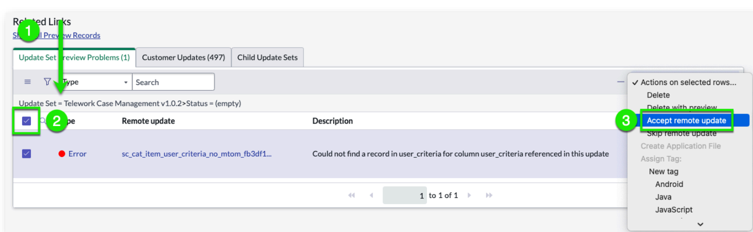
Update Set Super Guide

1. Scroll to the related lists

2. Select 'All' records under 'Update Set Preview Problems'

3. Expand 'Actions on selected rows' option box

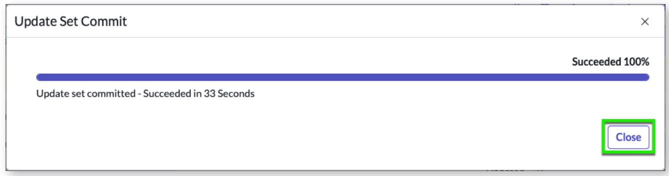
4. Click on 'Accept Remote Update'



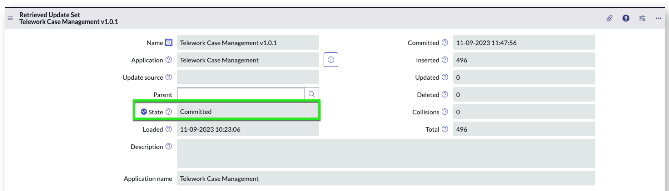

# Step 7 - Commit Update Set

1. Click on 'Commit Update Set'

2. Click on **Close**

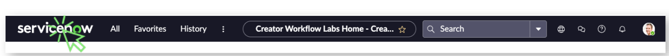



3. You should now see the application has been successfully imported:

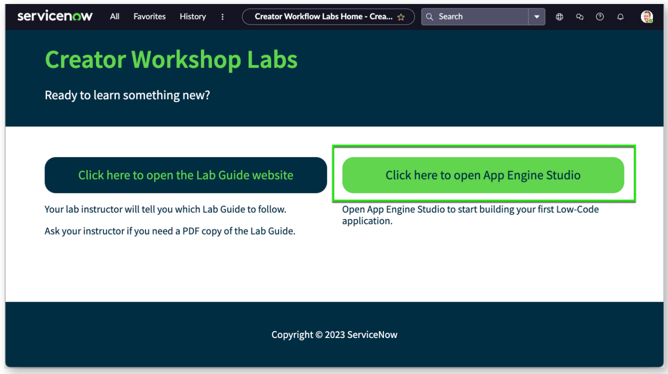



# Step 8 - Open App Engine Studio

1. Click on the ServiceNow logo to go to the home page




2. Click the button **Click here to open App Engine Studio**




Congratulations!

You have completed importing the Telework Case Management application.

Continue into App Engine Studio to find your application - Telework Case Management.

# Governance

## Use case

In the previous lab, we played the role of Amanda and built a fully functional Telework Case Management application to help the team efficiently manage work across the IT, HR and employee departments.

Amanda would like to add more features to her application before making it available to users for testing. However, as she is not sure how to implement them herself, it was recommended that she invites skilled collaborators to aid in the development process.

The application will contain sensitive employee information so she would like to make sure that regular platform administrators don't have access to the application.

Additionally, the platform admins want to make sure that the custom app developers can't create records on tables outside of their app.

We'll leverage the App Engine governance capabilities to invite collaborators and some platform security features to secure the app and the platform.

Let's get started.

# Overview

Amanda has successfully rolled out her application to her team, and it's caught the attention of another team who also wishes to use it. A primary condition is to maintain the privacy of records across the two teams.

In this exercise, we will compartamentalize telework cases for the users in the Development department and for the users in the Customer support team.

We'll configure this by defining:

> 1. One fulfiller role for each team: Fulfiller_Development and Fulfiller_Customer_Support

> 2. Then we'll define specific access controls for each.

> 3. In the end, one team will have visibility only over requests from users in the Development department while the other team will have access only to requests from the users in the Customer support department.

This will ensure both teams can use the app while maintaining the necessary separation and privacy of data.

Let's get started.

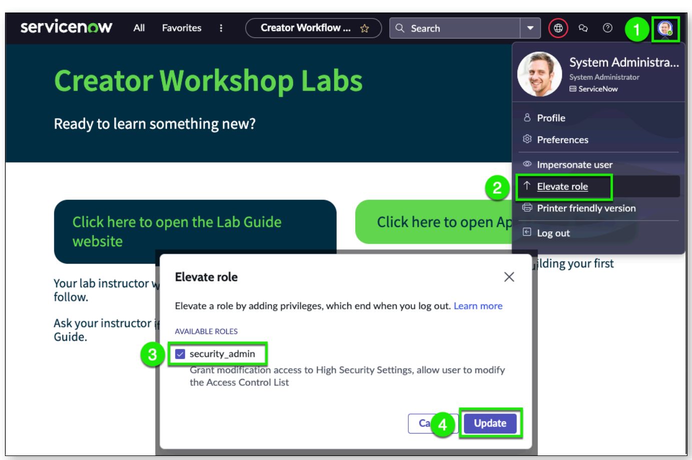
Citizen developers like Amanda need special privileges to configure access controls so Amanda is going to request...

## Help from App Engine Admins

1. Logged in as Admin

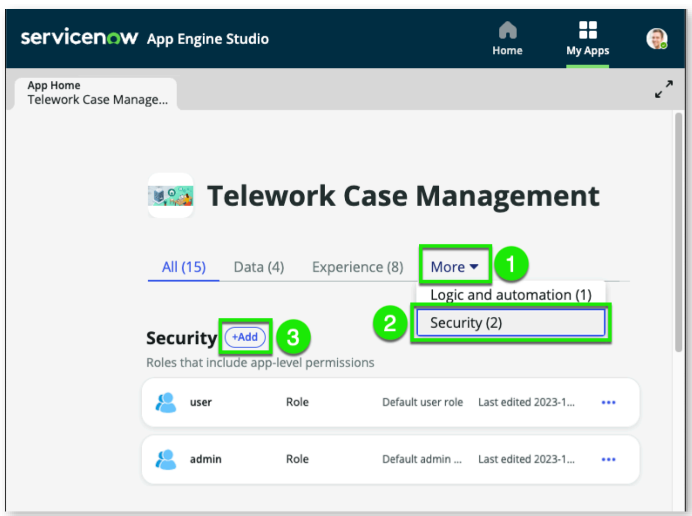
2. Elevate to Security Admin

| Steps | |
|---|---|
| 1 | In the top right, click the **Avatar** |
| 2 | Select **Elevate role** |
| 3 | Check [✔] security_admin |
| 4 | Click Update. |



3. Go to App Engine Studio > Security

| Steps | |
|---|---|
| 1 | In AES, click More |
| 2 | Click Security |
| 3 | Click Add |

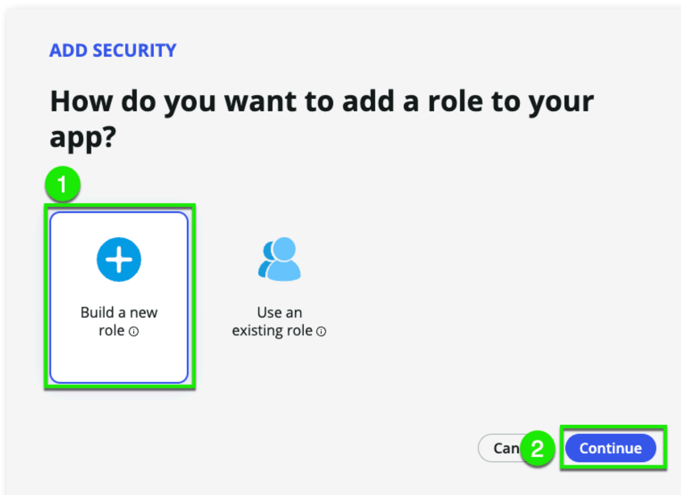



ⓘ INFO

When Amanda created the app it was only for one team and having only one user role was enough. Now that we need to open the application to another team we need to have more specific roles to ensure proper security.

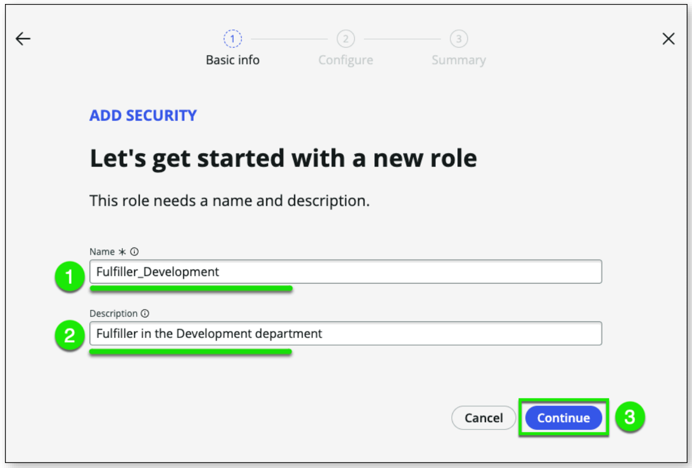
4. Create role Fulfiller_Development

| Steps | |
|---|---|
| 1 | Click build a new role |
| 2 | Click Continue. |

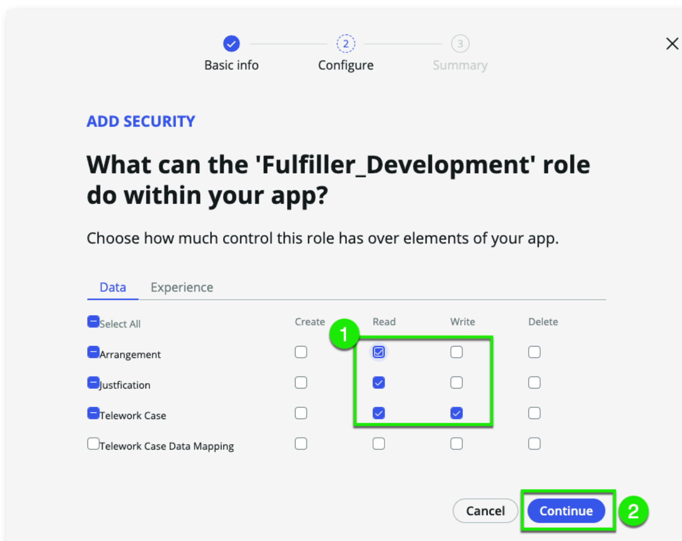



5. Give the role a name and description.

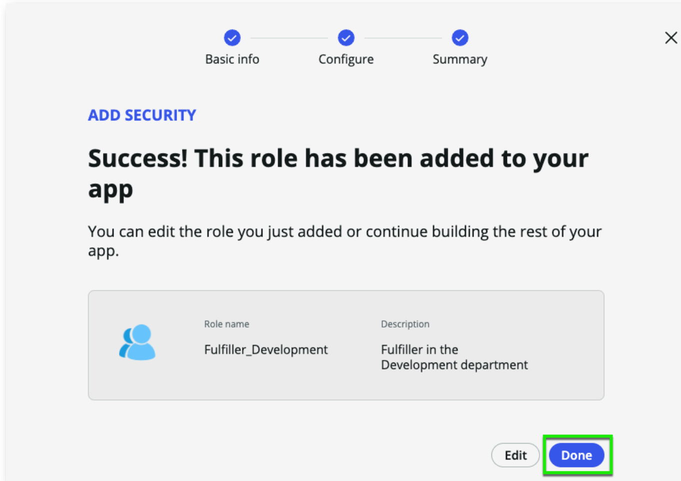
| Steps | |
|---|---|
| Name | Fulfiller_Development |
| Description | Fulfiller in the Development department |




6. Give the role the desired permissions



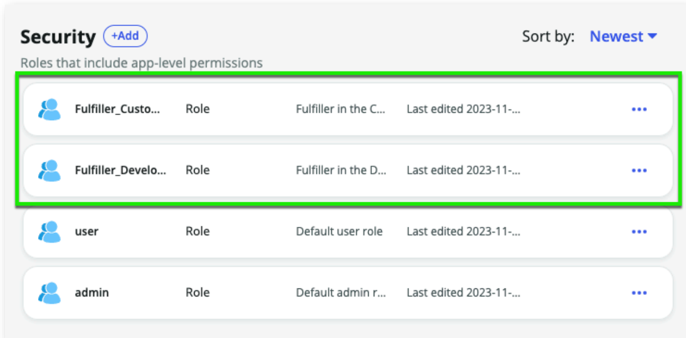

7. Success
Click Done.




7. Repeat the steps 4 to 7 to create the role **Fulfiller_Customer_Support**

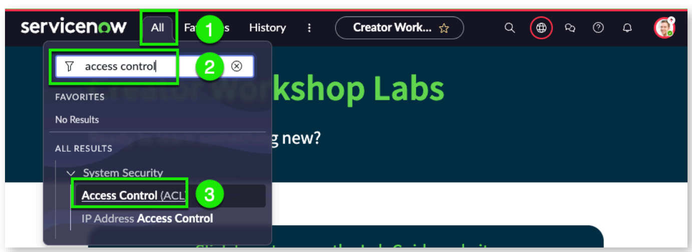
| Steps | |
|---|---|
| Name | Fulfiller_Customer_Support |
| Description | Fulfiller in the Customer Support department |

8. You should see the two new roles in the app




## Refine the access controls

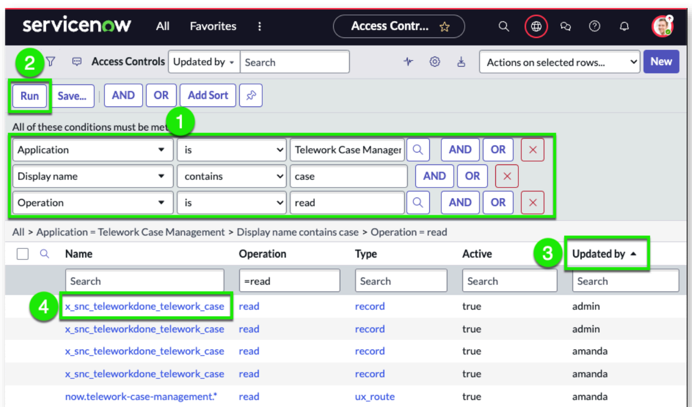
1. Navigate to the Access Control List (ACL)




2. Filter the list of records

| Filter | |
|---|---|
| 1 | Set the filter as shown below |

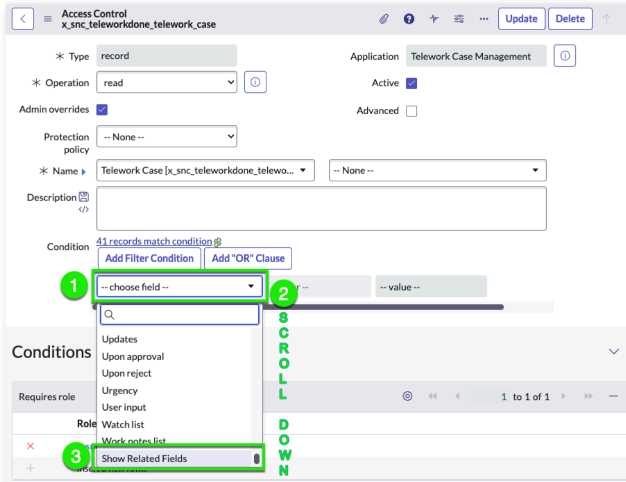
| Filter | |
|---|---|
| 2 | Click Run |
| 3 | Click on the **Updated by** to get the admin ACL at the top |
| 4 | Open the first record |




3. Review the ACL

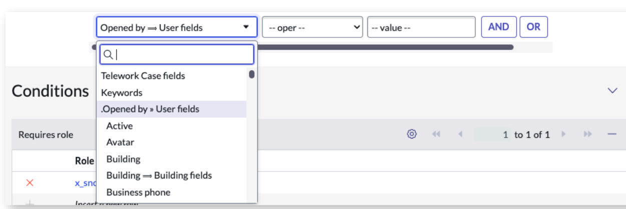
4. Define the condition: Opened by someone in the Development department

    1. We click on the condition field, scroll down to select **Show related fields**

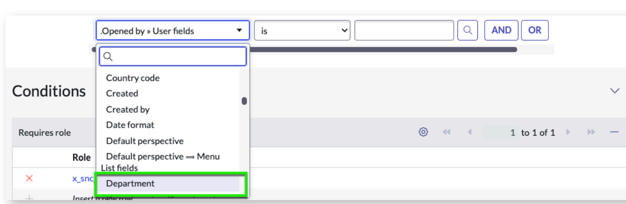



2. Select "Opened by ⇨ User fields"

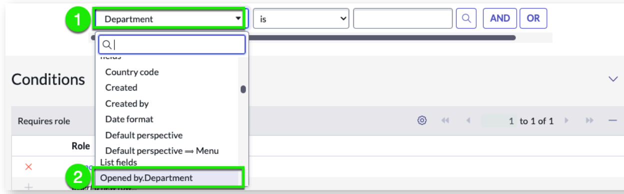

3. Click again on the field and we can now see the fields inside the Opened by



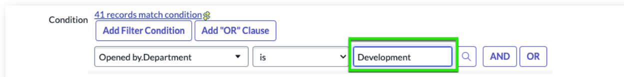

4. Scroll down and select **department**



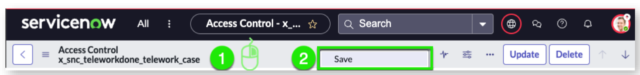
5. Click again on the field and then select **Opened by Department**



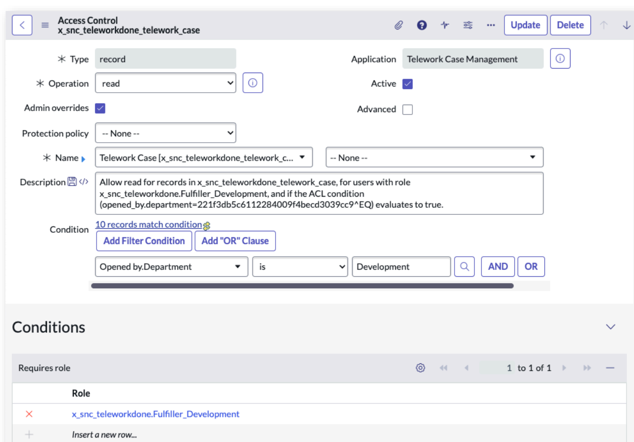

6. Finally set the condition value to **Development**



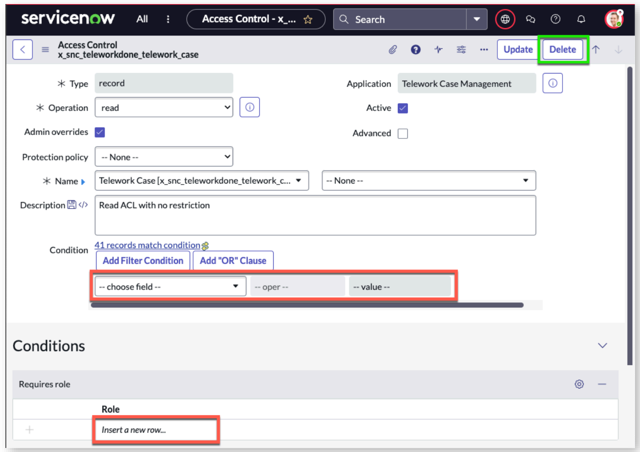

7. Save the record




8. Notice ServiceNow has generated a description




9. To secure the data access let's review the other ACL our work
We need to delete any ACL that does not have any restriction

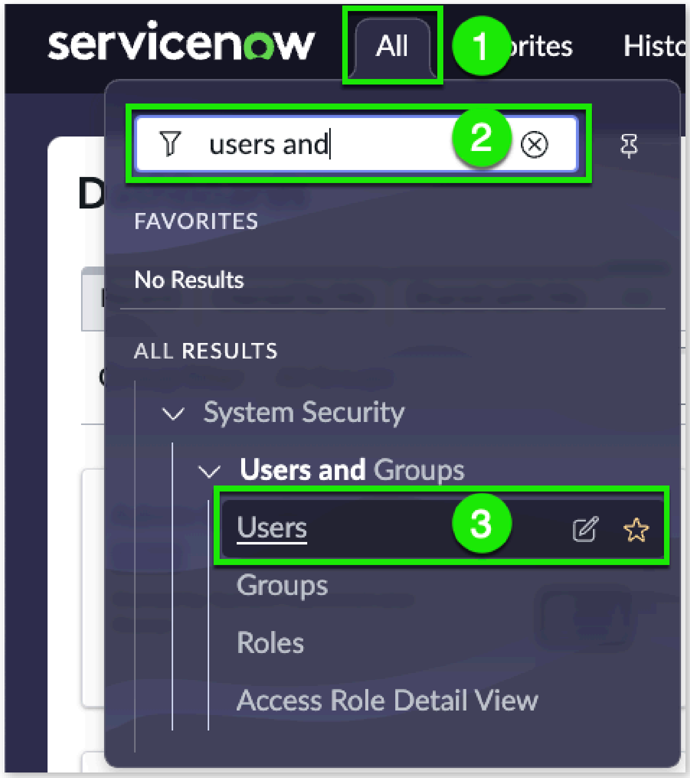



## Assign the role to Abel

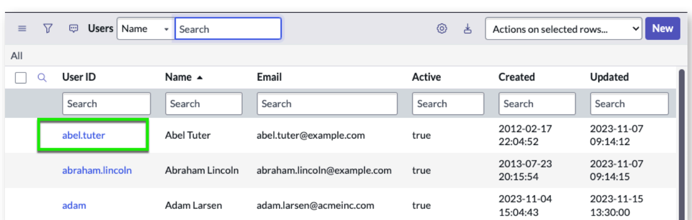
### Open user Abel

1. Go to the list of users

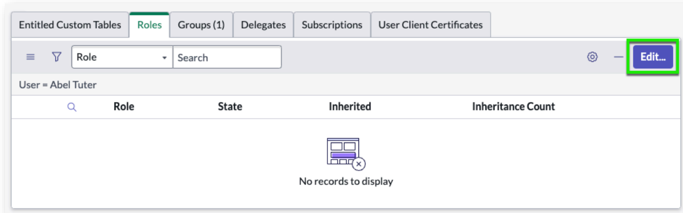
| Steps | |
|---|---|
| 1 | Click the **All** menu |
| 2 | Search for **Users** |
| 3 | Click on **Users** |



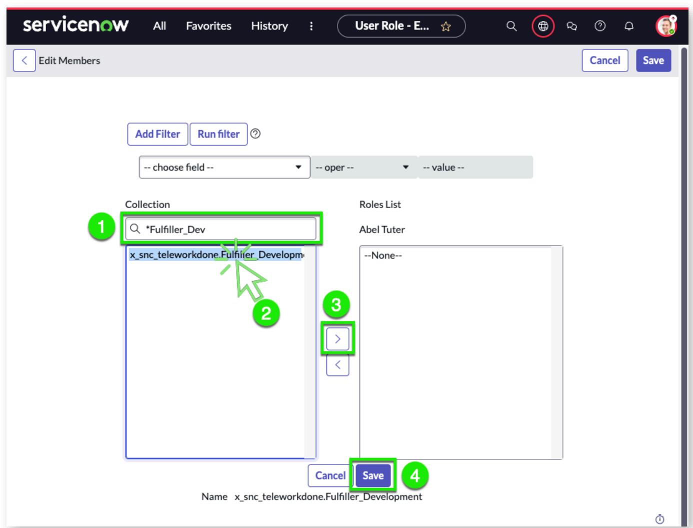

2. Open the user Abel




3. Click Edit Roles



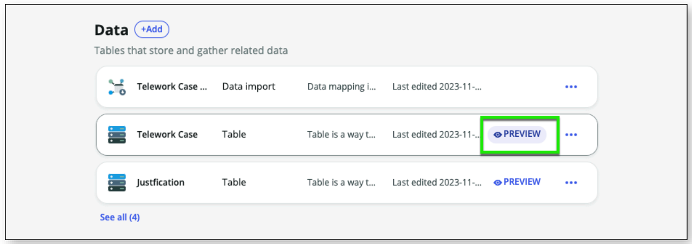

4. Grant Abel the telework user role



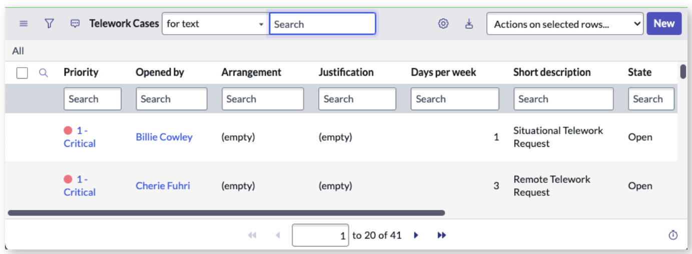

## Let's test

1. Go to App Engine Studio

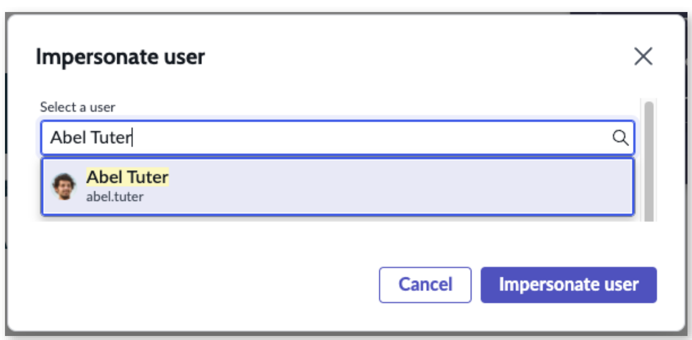
2. Preview the Telework case table



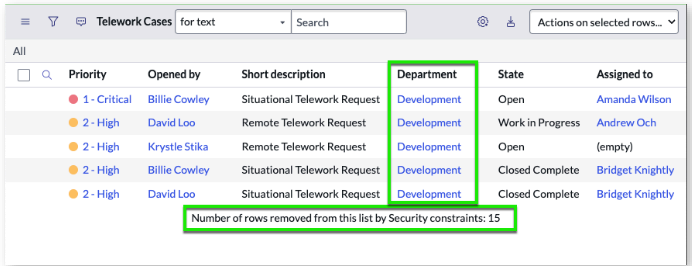

3. Notice we can see all the records




4. Go back to the admin view & Impersonate Abel




5. Go back to the Telework case list and Notice we can see all the records



Telework Cases for text Search Actions on selected rows...

All

| Priority | Opened by | Short description | Department | State | Assigned to |
|---|---|---|---|---|---|
| 1 - Critical | Billie Cowley | Situational Telework Request | Development | Open | Amanda Wilson |
| 2 - High | David Loo | Remote Telework Request | Development | Work in Progress | Andrew Och |
| 2 - High | Krystle Stika | Remote Telework Request | Development | Open | (empty) |
| 2 - High | Billie Cowley | Situational Telework Request | Development | Closed Complete | Bridget Knightly |
| 2 - High | David Loo | Situational Telework Request | Development | Closed Complete | Bridget Knightly |

Number of rows removed from this list by Security constraints: 15

# Flow: Overview

### Purpose

In this exercise, we will learn how to implemenent an integration using Spoke Generator, how to create a Decision Table and how to leverage them in our main flow to add straigh-through proccesing.

Amanda has learned the finance department has pre-certified employees for telework. In the case of transfers, requests for telework should be automatically approved. This data has not been transferred to the ServiceNow instance, but the finance department has published an OpenAPI that can be used to look-up employee data based on their email address.

Managers have also informed Amanda that when employees apply for 3 days or less of situational telework, the requests can be automatically approved.

### In this part of the exercise, we will:

- Create a re-usable integration to the finance department using Spoke Generator.
- Create a Decision Table to check if the telework request can be automatically approved.
- Modify our Main Flow to by-pass manager approval if the employee was pre-certified for telework.

# Notes

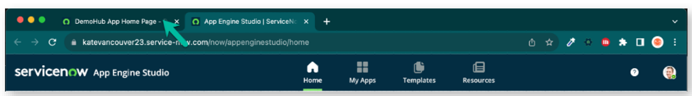
-

# Create People Finder Spoke

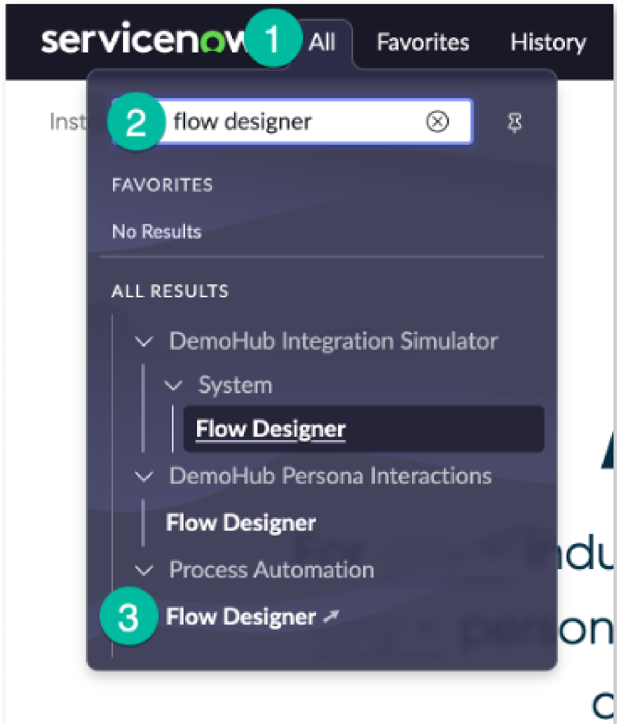
We will create a re-usble integration to the finance department's People Finder Open API, so that we can access this data in our Main flow.

## Open Flow Designer to use Spoke Generator to create the People Finder Integration

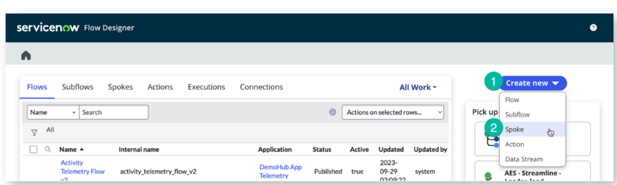
1. Go to the tab with the ServiceNow Admin Home Page (your title may be different).




2. Open Flow Desginer from the All Menu
    1. Click the **All** menu
    2. Search for "flow designer"
    3. Click on: Process Automation > Flow Designer

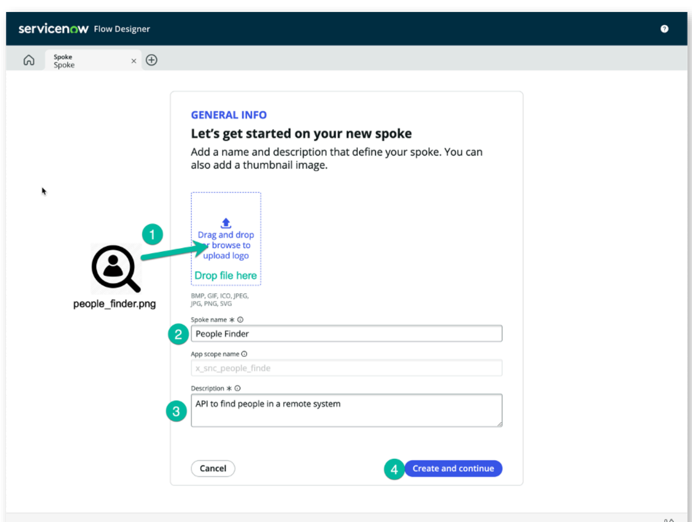



3. Create New Spoke
    1. Click Create new ▼
    2. Click "Spoke"

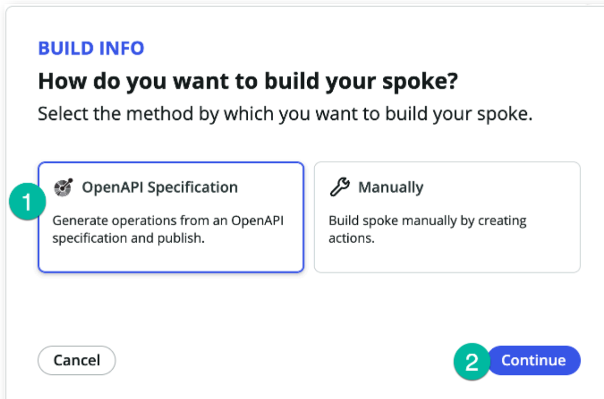


4. Setup General Information for the Spoke (💡 double-click to select, copy & paste into the form).

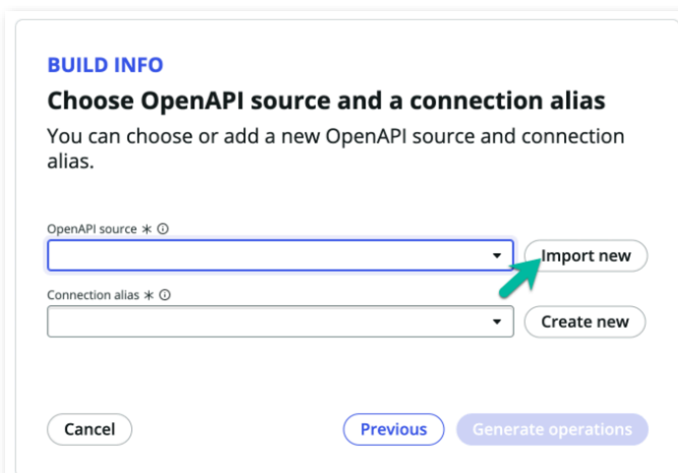
| | Field Name | Field Value |
|---|---|---|
| 1 | Drag and drop the icon in the telework_lab files folder we downloaded earlier | people_finder.png |
| 2 | Spoke Name (App scope name will auto-populate) | People Finder |
| 3 | Description | API to find people in a remote system |
| 4 | Click on the Create and continue button. | |



5. Specify how we want to build our Spoke
    1. Select OpenAPI Specification
    2. Click Continue

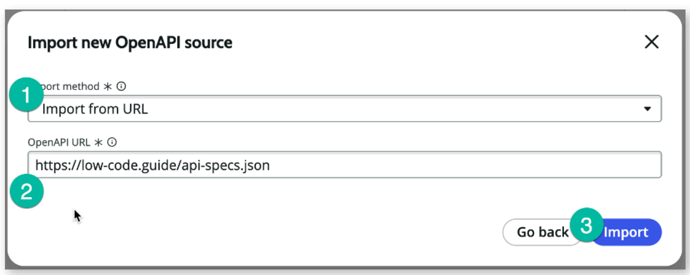



6. For the OpenAPI source, click Import new

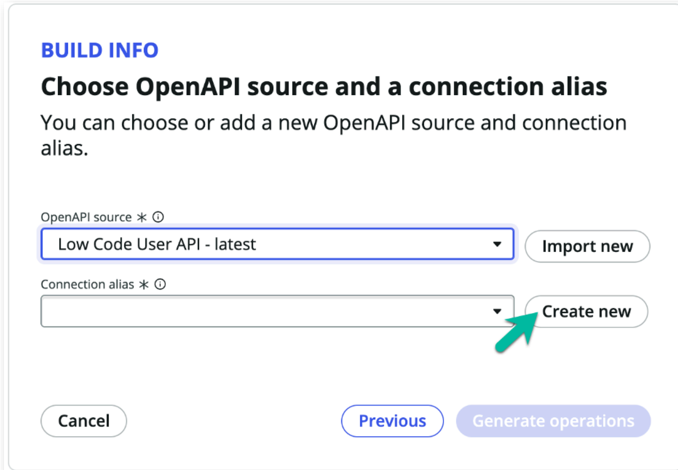


7. Fill the user form with the following values ( 💡 double-click to select, copy & paste into the form).

| | Field Name | Field Value |
|---|---|---|
| 1 | Import method | Import from URL |
| 2 | OpenAPI URL | https://low-code.guide/api-specs.json |
| 3 | Click on the Import button. | |

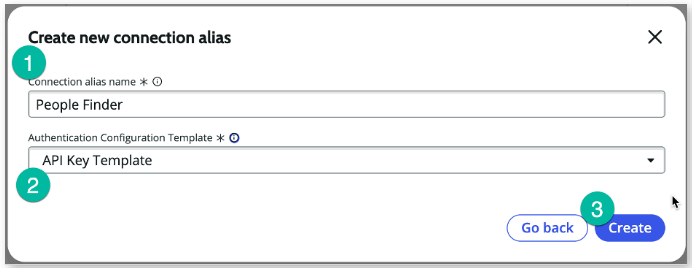



8. For the Connection alias, click Create new

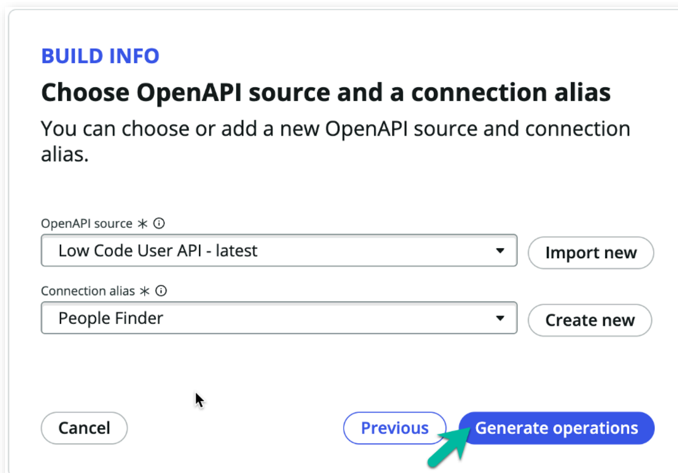



9. Fill the user form with the following values (💡 double-click to select, copy & paste into the form).

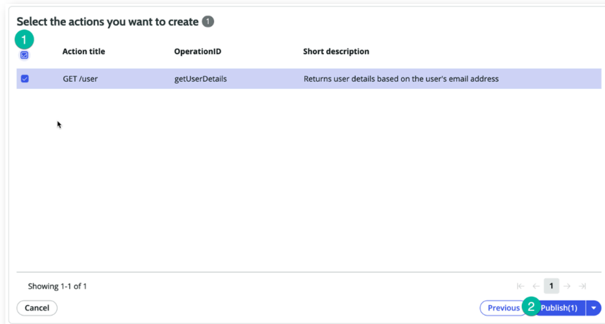
| | Field Name | Field Value |
|---|---|---|
| 1 | Connection alias name | People Finder |
| 2 | Authentication Configuration Template | API Key Template |
| 3 | Click on the Create button. | |



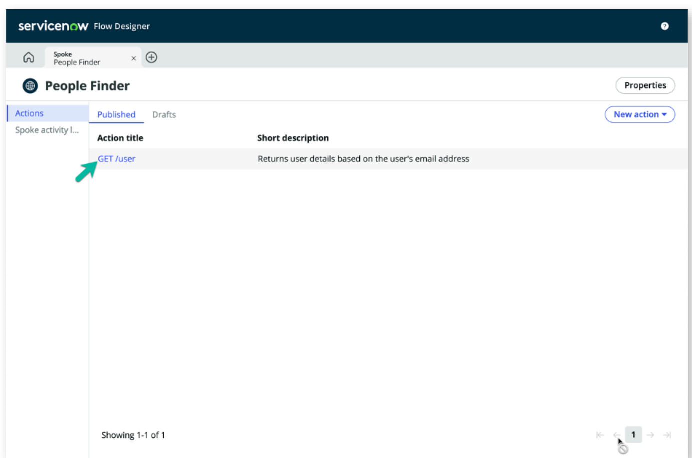

10. To continue, click the Generate operations button




11. Select the actions you want to create
    1. Select all actions
    2. Click the Publish(1) | ▼ button

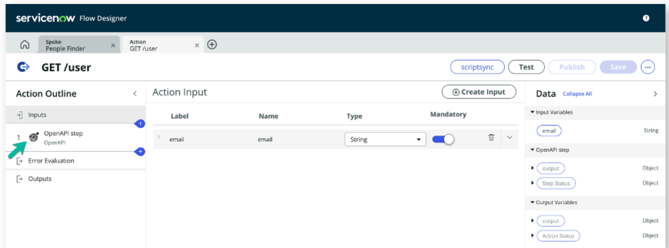



12. Click the "GET /user" link to view the action we just created

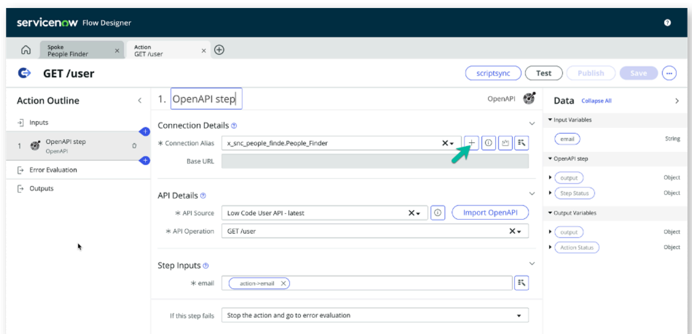



13. Before we can test and use our spoke, we need to set the Base URL for the Connection.

> Click to expand the OpenAPI step



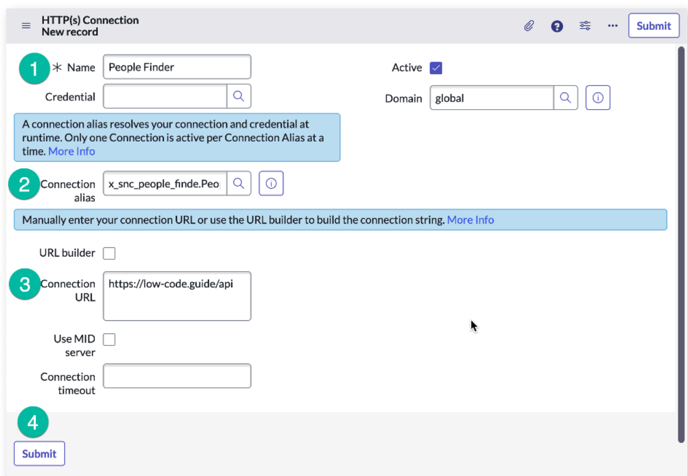

14. Click the plus sign next to the Connection Alias




15. Set up the HTTPs(s) Connection ( double-click to select, copy & paste into the form).

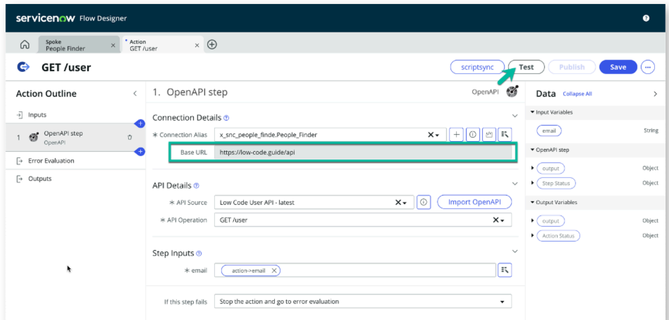
| | Field Name | Field Value |
|---|---|---|
| 1 | Name | People Finder |
| 2 | Connection alias | x_snc_people_finde.People_Finder |

| | Field Name | Field Value |
|---|---|---|
| 3 | Connection URL | https://low-code.guide/api |
| 4 | Click on the Submit button. | |



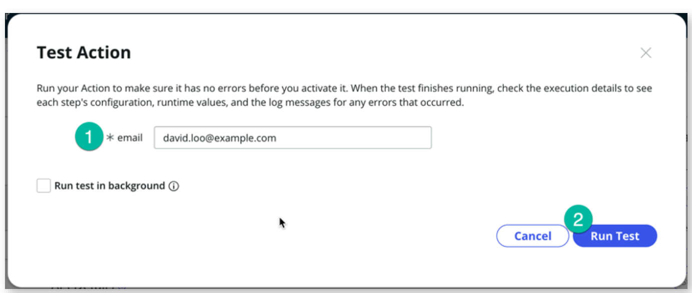

16. The Base URL should now be populated.

    Click on the Test button

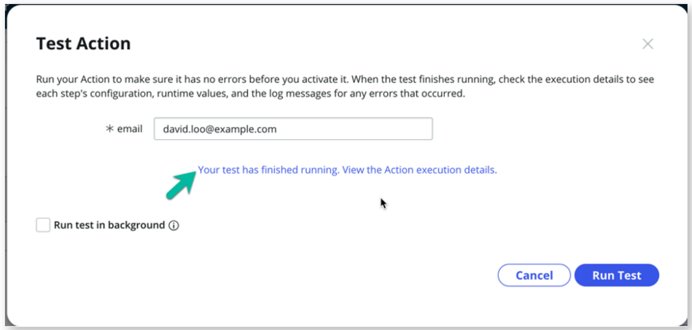



17. Test the action with David Loo's email (💡 double-click to select, copy & paste into the form).

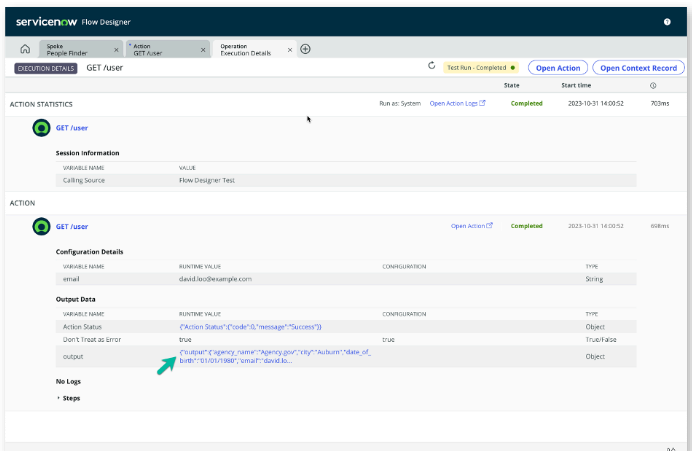
| | Field Name | Field Value |
|---|---|---|
| 1 | email | david.loo@example.com |
| 2 | Click on the Run Test button. | |




18. When the test completes, click the link to view the execution results

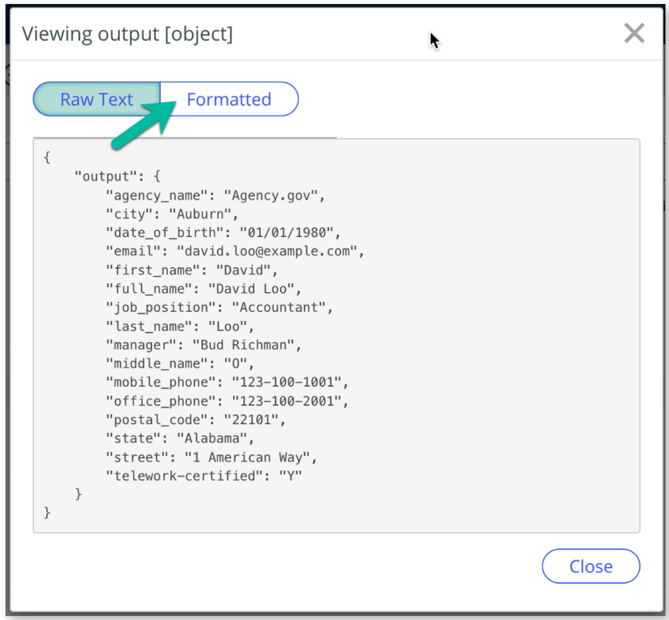



19. Click on the RUNTIME VALUE link

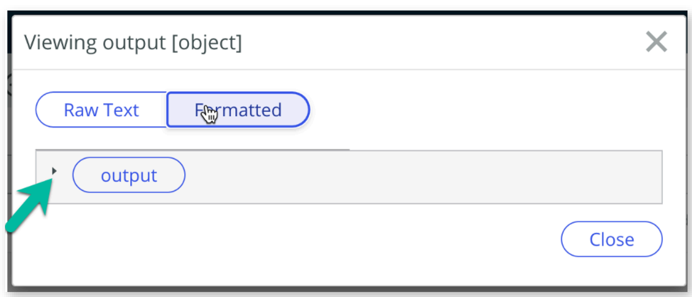


20. Review the JSON results. We can switch to the Formatted view.

> Click Formatted toggle.

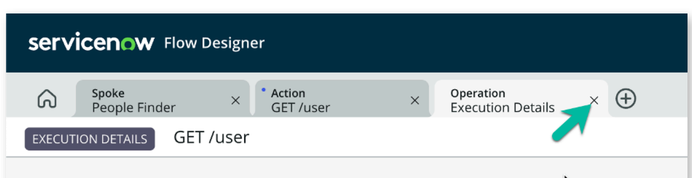



21. Expand the output.

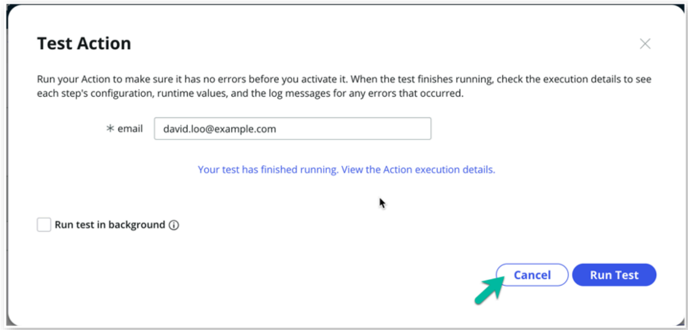



22. Review the Formatted JSON results.
    1. Note the **telework-certified** value.
    2. Click Close


23. Close the Execution Details tab

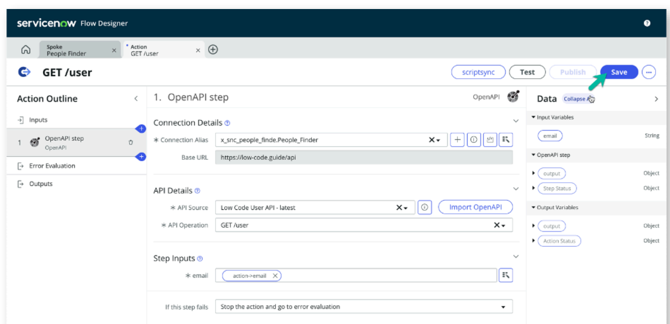



24. Close the Test Action model by clicking the Cancel button



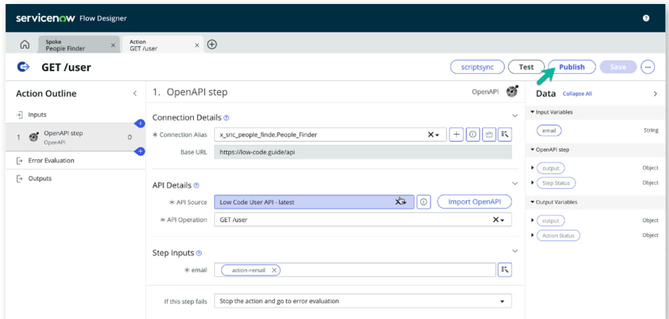

25. Save the GET /user spoke action

Click the Save button



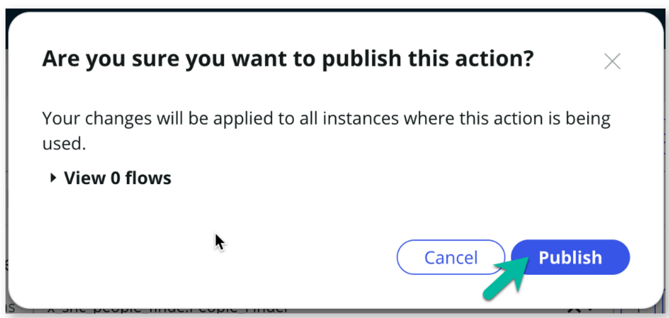

26. Now, we publish the spoke, so it will be available to call from our Main Flow.

Click the Publish button




27. Confirm publication of the action

Click the Publish button

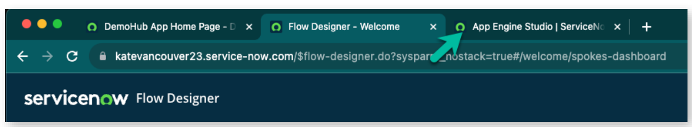


# Create Decision Table for Auto Approval Conditions

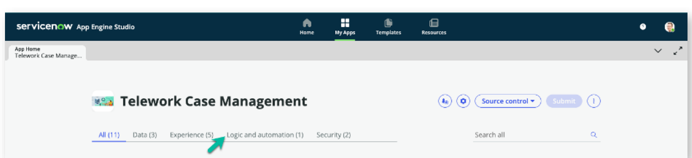
We will create a decision table to assess the arrangement type and numbers of days and return an auto-approval result that we can access in our Main flow.

## Return to App Engine Studio to Build the Decision Table

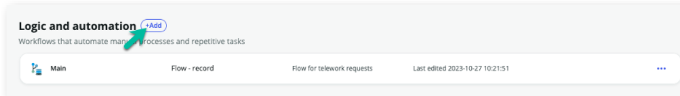
1. Go to the tab with the App Engine Studio.



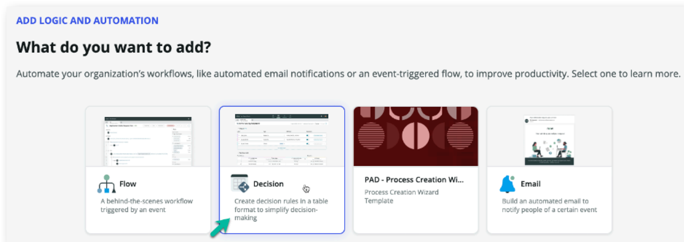

2. Click on "Logic and Automation (1)".



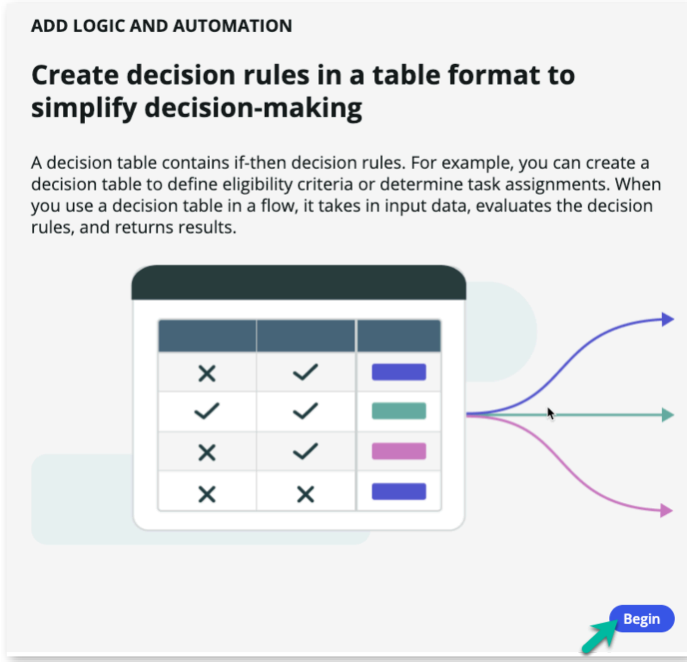

3. Click the +Add button.




4. Select the Decision tile.

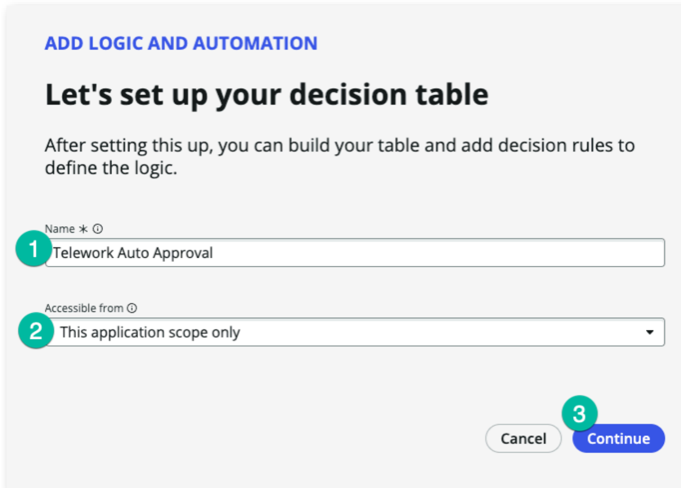



5. Click the Begin button.

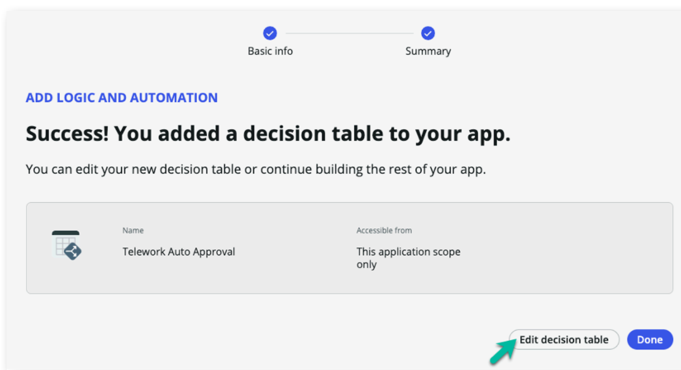



6. Set up your decision table.

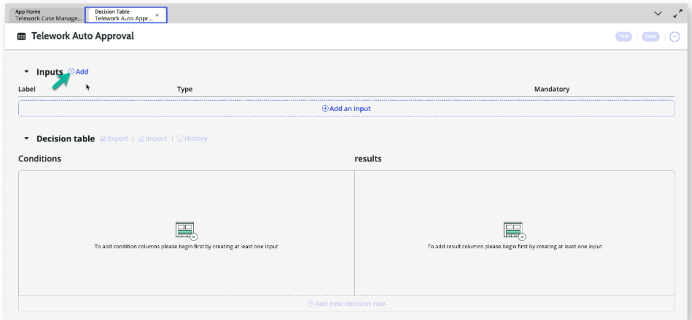
| | Field Name | Field Value |
|---|---|---|
| 1 | Name | Telework Auto Approval |
| 2 | Accessible from | This application scope only |
| 3 | Click on the Continue button. | |




7. Click the Edit Decision Table button.

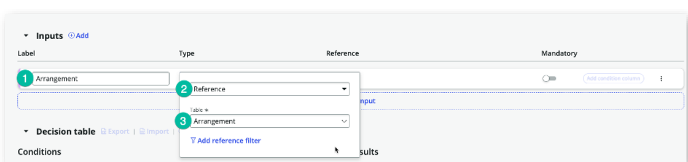



8. Click the ◇ Add button next to Inputs.

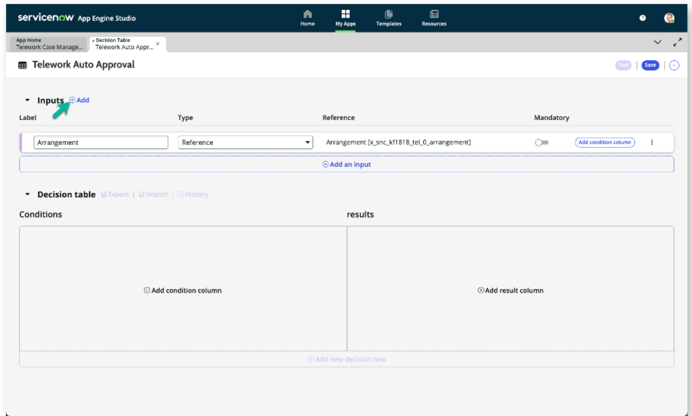



9. Populate the following:

| | Field Name | Field Value |
|---|---|---|
| 1 | Label | Arrangement |
| 2 | Type | Reference |
| 3 | Table | Arrangement |

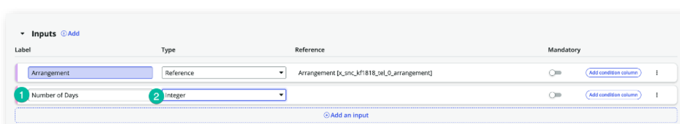



10. Click the Add button next to Inputs again.




11. Populate the following:

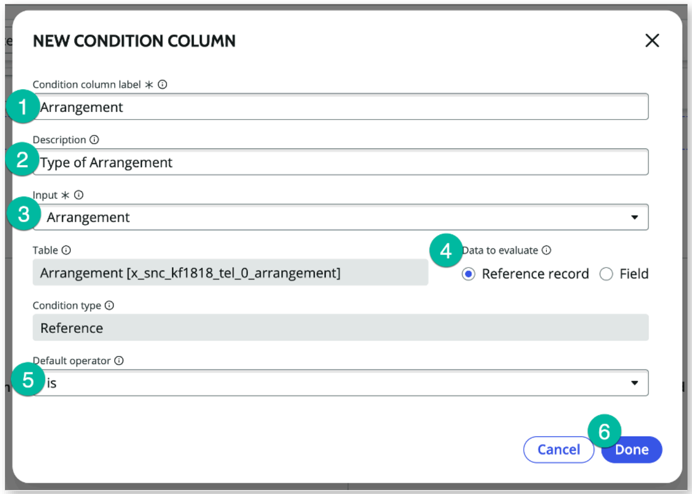
| | Field Name | Field Value |
|---|---|---|
| 1 | Label | Number of Days |
| 2 | Type | Integer |




12. In the Decision Table section, click Add condition column.


13. Populate the NEW CONDITION COLUMN form.

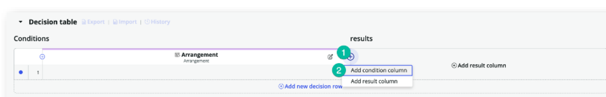
| | Field Name | Field Value |
|---|---|---|
| 1 | Condition Column Label | Arrangement |
| 2 | Description | Type of Arrangement |
| 3 | Input | Arrangement |
| 4 | Data to Evaluate | Reference Record |
| 5 | Default operator | is |
| 6 | Click on the Done button. | |




14. Add another condition column.

    1. Click the plus button ⊕ next the condition we just created.

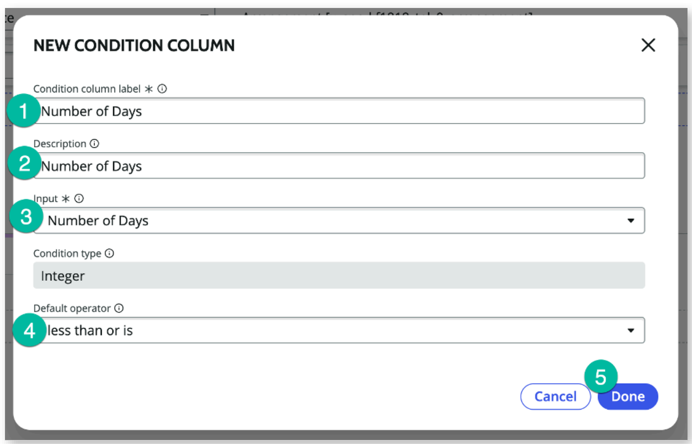
    2. Select "Add condition column".



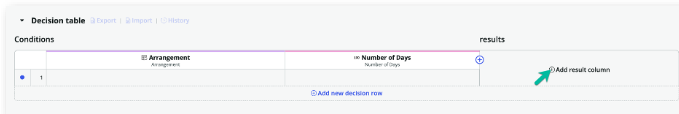

15. Populate the NEW CONDITION COLUMN form.

| | Field Name | Field Value |
|---|---|---|
| 1 | Condition Column Label | Number of Days |

| | Field Name | Field Value |
|---|---|---|
| 2 | Description | Number of Days |
| 3 | Input | Number of Days |
| 4 | Default operator | less than or is |
| 5 | Click on the Done button. | |

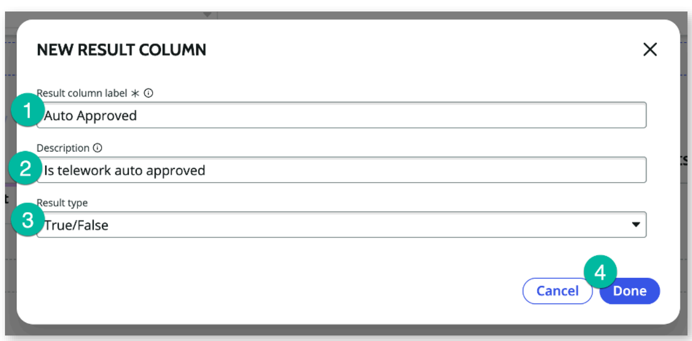



16. In the results column, click Add result column.

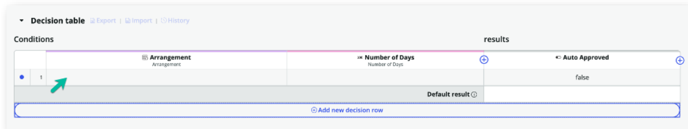



17. Populate the NEW RESULT COLUMN form.

| | Field Name | Field Value |
|---|---|---|
| 1 | Result Column Label | Auto Approved |
| 2 | Description | Is telework auto approved |
| 3 | Result Type | True/False |
| 4 | Click on the Done button. | |

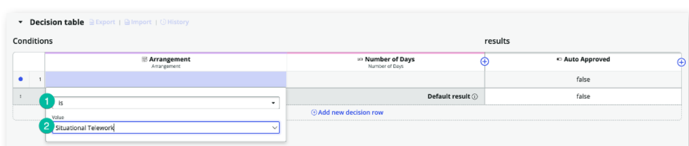


18. Let's define our first decision row, by clicking into row 1.




19. Populate the Arrangement Condition.

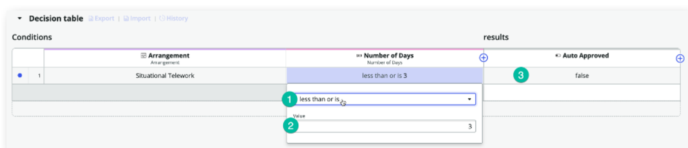
| | Field Name | Field Value |
|---|---|---|
| 1 | Operator | is |
| 2 | Value | Situational Telework |



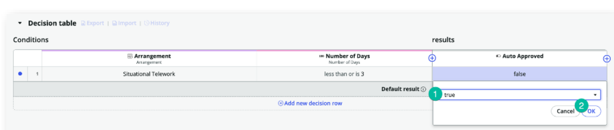

20. Populate the Number of Days Condition.

| | Field Name | Field Value |
|---|---|---|
| 1 | Operator | less than |

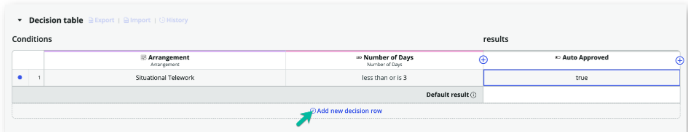
| | Field Name | Field Value |
|---|---|---|
| | | or is |
| 2 | Value | 3 |
| 3 | Click on the value in the Auto Approved column (currently false). | |




21. Change the result for Auto Approval to true.
    1. Switch the result to true.
    2. Click the OK button to continue.

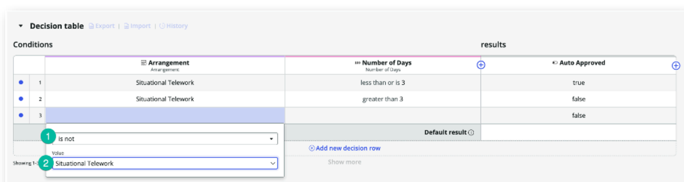



22. Let's add our next decicion row by clicking Add new deicion row




23. Populate the Arrangement Condition.

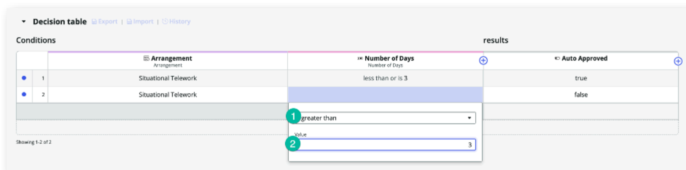
| | Field Name | Field Value |
|---|---|---|
| 1 | Operator | is |
| 2 | Value | Situational Telework |



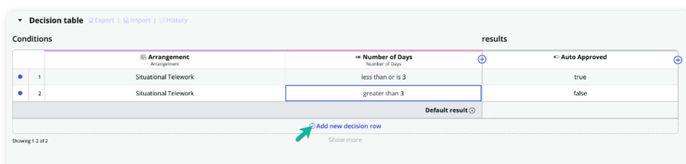

24. Populate the Number of Days Condition.

| | Field Name | Field Value |
|---|---|---|
| 1 | Operator | greater than |

| | Field Name | Field Value |
|---|---|---|
| 2 | Value | 3 |

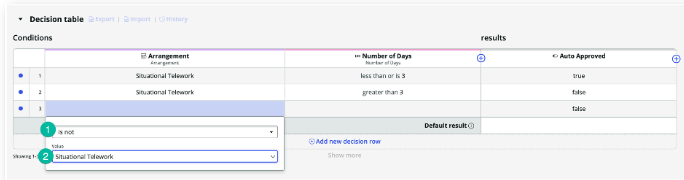



25. Let's add another decicion row by clicking Add new deicion row

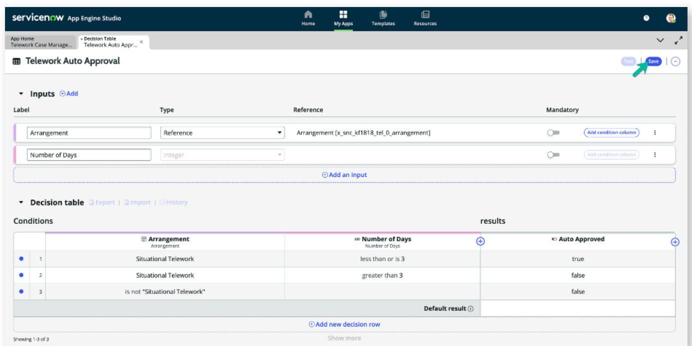



26. Populate the Arrangement Condition.

| | Field Name | Field Value |
|---|---|---|
| 1 | Operator | is not |
| 2 | Value | Situational Telework |



27. Let's save our work by clikcing the Save button.



## Bonus

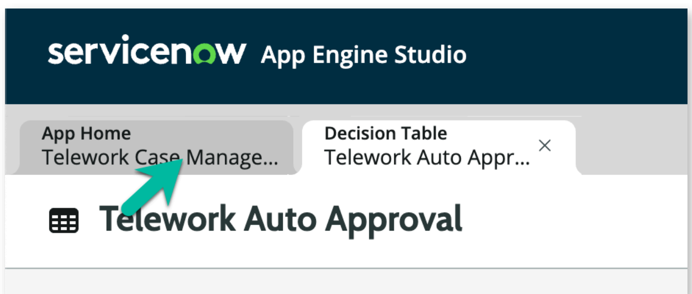
Test diffent scenarios before modifying the Main Flow.

# Modify the Telework Case Main Flow

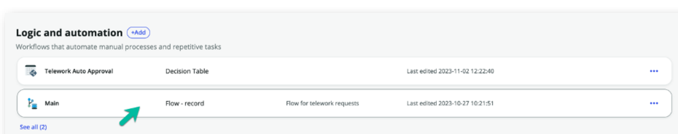
Now that we have created the People Finder Spoke and our Auto Approval Decision Table, let's modify the Telework Case Main Flow to allow auto-approval for those that were pre-certified or have requested 3 days or less for Situational Telework.

## Open the Main Flow and add our changes

1. Return to the App Home tab in App Engine Studio.

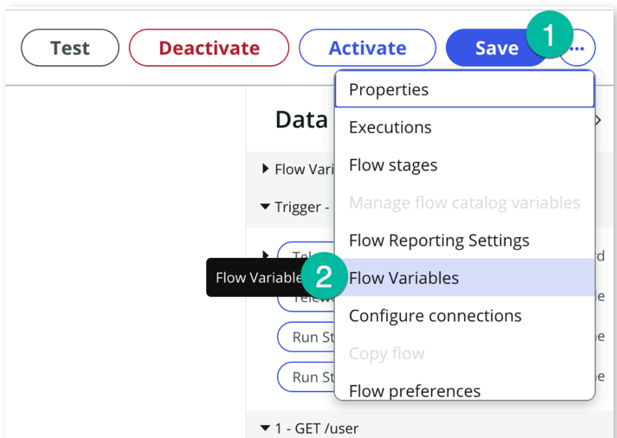



2. Open the Main flow by clicking on the row under Logic and Automation.




3. Toggle the Flow Designer view to Tree View.

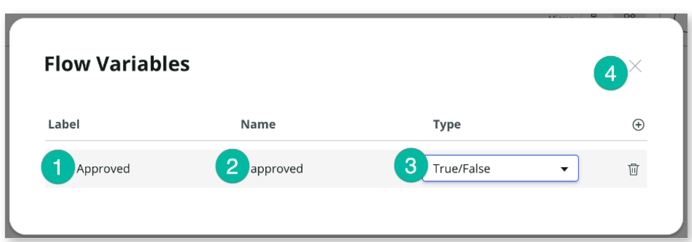
4. First, we want to add a Flow Variable to store the status of auto approval.
   1. Click the ... button next to the Save button on the top-left of your screen.
   2. Select "Flow Variables" from the drop-down.



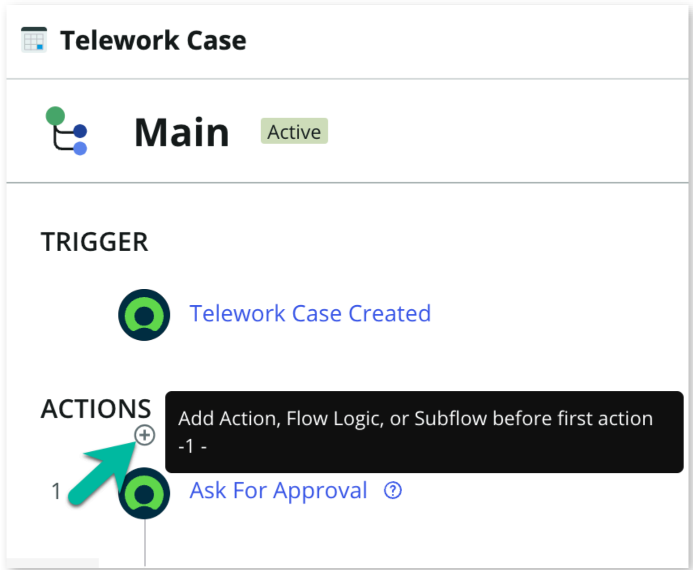

5. Click the plus sign ⊕ to add a variable.


6. Set up the flow variable.

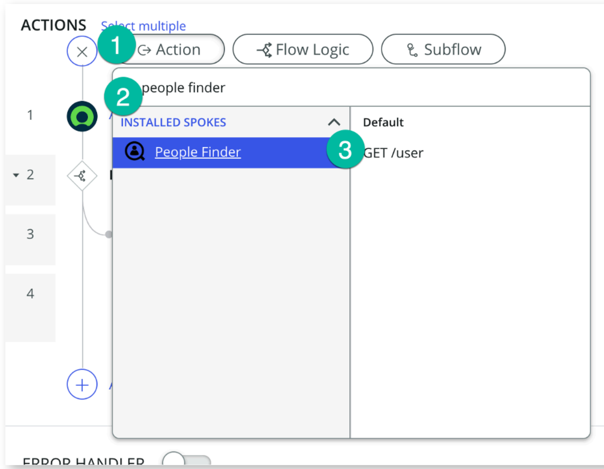
| | Field Name | Field Value |
|---|---|---|
| 1 | Label | Approved |
| 2 | Name | approved |
| 3 | Type | True/False |
| 4 | Click on the X button to close. | |



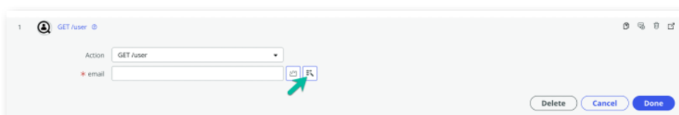
7. Before our first step, we want to call the People Finder Spoke. Hover above the "Ask for Approval" step and click the plus sign ⊕.



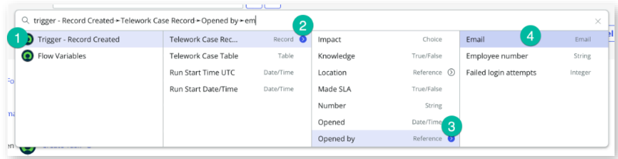

8. Add the People Finder Spoke
    1. Select "Action" as the step type.
    2. Search for "people finder".
    3. Select the "GET /user" action.



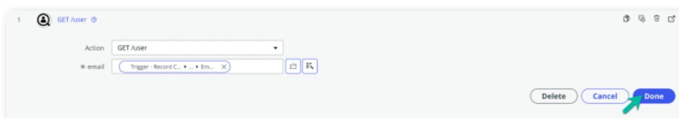

9. Click the data pill picker for email.



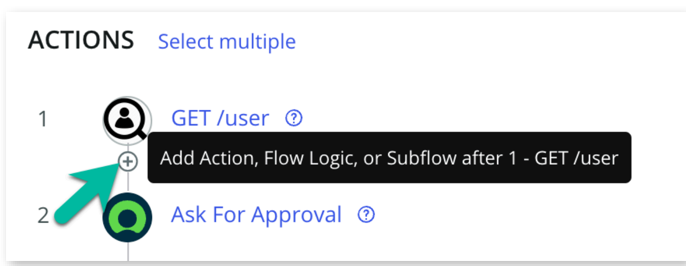

10. Dot-walk to the email address of the person that opened the request.
    1. Click "Trigger - Record Ceated"
    2. Click the blue arrow next to the "Telework Case" record to drill down.
    3. Click the blue arrow next to "Opened by".
    4. Click on "Email" to set the value.



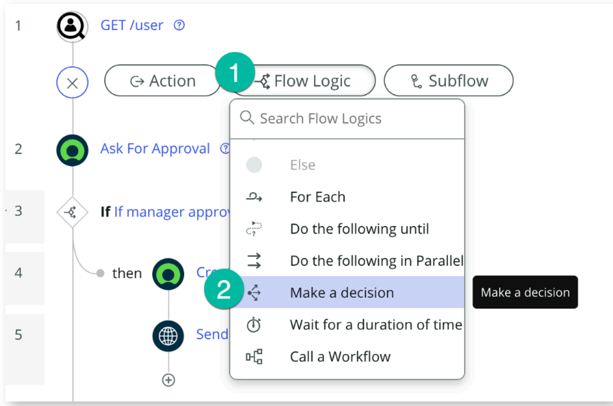

11. Click the Done button.




12. Next, we want to use the decision table we built. Hover in-between steps 1 and 2 and click the plus sign ⊕.



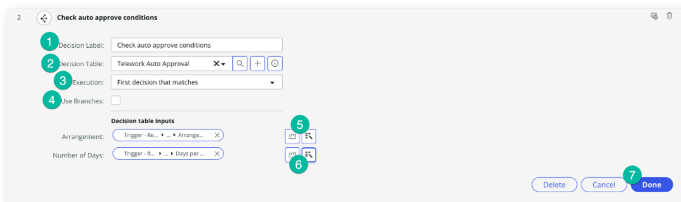

13. Add the Telework Auto Approval Decision Table.
    1. Select "Flow Logic" as the step type.
    2. Select "Make a decision" from the list.




14. Setup the decision step.

| | Field Name | Field Value |
|---|---|---|
| 1 | Decision Label | Check auto approve conditions |
| 2 | Decision Table | Telework Auto Approval |
| 3 | Execution | First decision that matches |
| 4 | Use Branches | Un-check |
| 5 | Arrangement | Dot-walk (see step 15 below) |
| 6 | Number of Days | Dot-walk (see step 16 below) |

| | Field Name | Field Value |
|---|---|---|
| 7 | Click on the Done button. | |

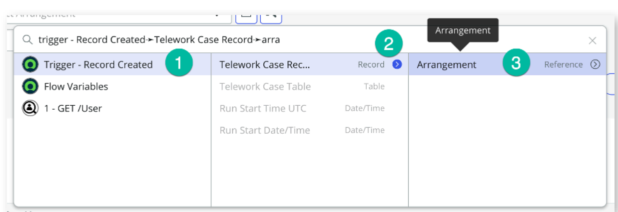



15. Dot-walk Arrangement (step 5 above)

    1. Click "Trigger - Record Ceated"

    2. Click the blue arrow next to the "Telework Case" record to drill down.

    3. Click on "Arrangement" to set the value.

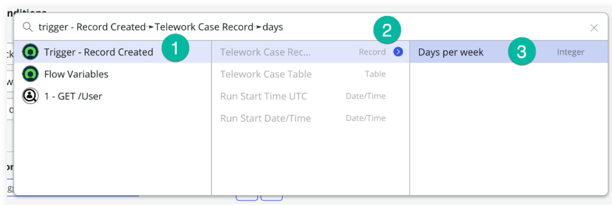



16. Dot-walk Number of Days (step 6 above)

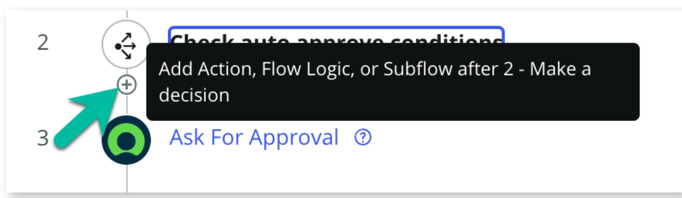
    1. Click "Trigger - Record Ceated"

2. Click the blue arrow next to the "Telework Case" record to drill down.

3. Click on "Days per week" to set the value.



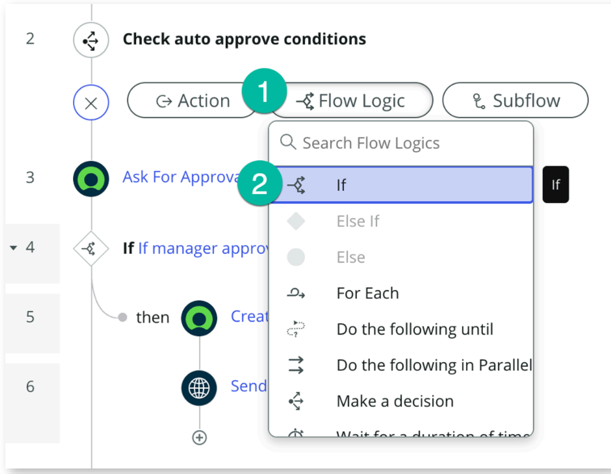

17. Now, we have to check if either auto-approve condition has been met. Hover in-between steps 2 and 3 and click the plus sign ⊕.




18. Add the if condition.

    1. Select "Flow Logic" as the step type.

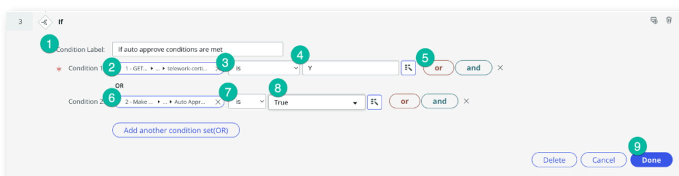
    2. Select "If" from the list.




19. Setup the if condition.

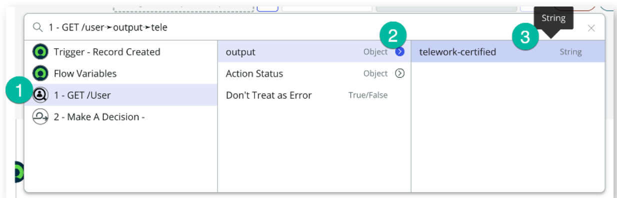
| | Field Name | Field Value |
|---|---|---|
| 1 | Condition Label | If auto approve conditions are met |
| 2 | Condition 1 | Dot-walk (see step 20 below) |
| 3 | Operator | is |
| 4 | Value | Y |
| 5 | Click on the or button. | |

| | Field Name | Field Value |
|---|---|---|
| 6 | Condition 2 | Dot-walk (see step 21 below) |
| 7 | Operator | is |
| 8 | Value | True |
| 9 | Click on the Done button. | |




20. Dot-walk Condition 1 (step 2 above)

    1. Click "1 - GET /User"
    2. Click the blue arrow next to the "output" object to drill down.
    3. Click on "telework-certified" to set the value.



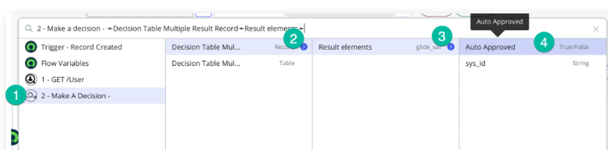

21. Dot-walk Condition 2 (step 2 above)

    1. Click "2 - Make A Decision"

    2. Click the blue arrow next to the "Decision Table Multiple Result Record" record to drill down.

    3. Click the blue arrow next to the "Result elements" glide_var to drill down.

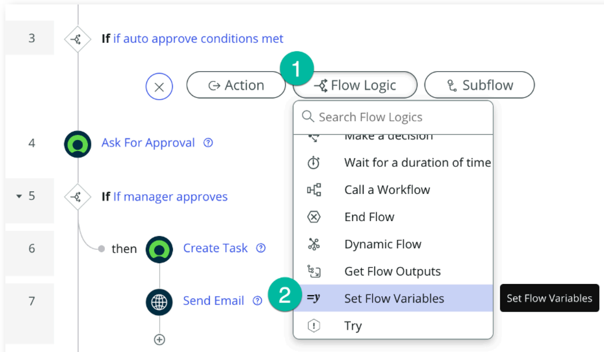
    4. Click on "Auto Approved" to set the value.



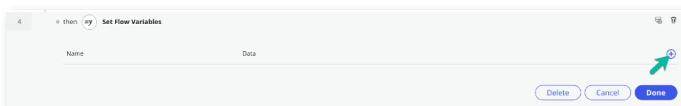

22. Let's set the Approved value to true.

    1. Select "Flow Logic" as the step type.

    2. Select "Set Flow Variables" from the list.

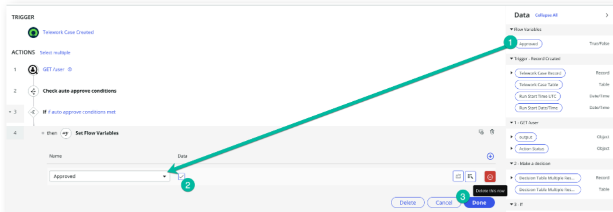



23. To add a flow variable to set, click the plus sign .

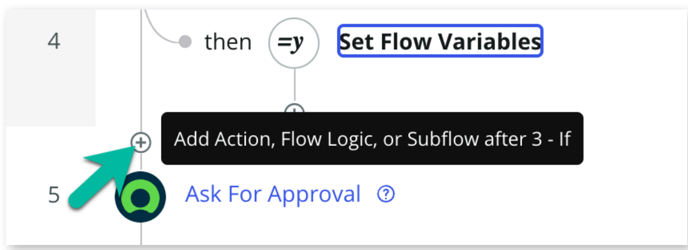



24. Set the Approved flow variable to true.

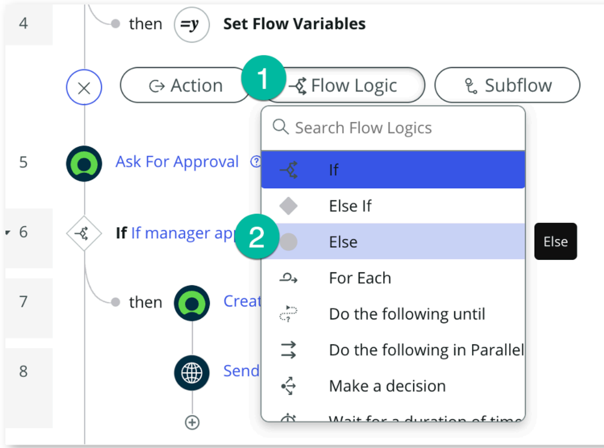
    1. Drag and drop the Approved flow variable from the data pills on the right side of the screen into the Name field.
    2. Check the data box for true.
    3. Click the "Done" button.



25. Now let's the flow logic for when our condition is not met. Hover above step 5 and click the plus sign ⊕.



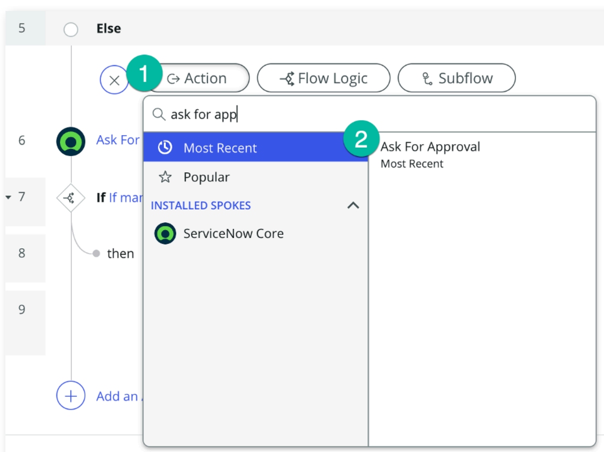

26. Add the else condition.
    1. Select "Flow Logic" as the step type.
    2. Select "Else" from the list.




27. Add the ask for approval action.

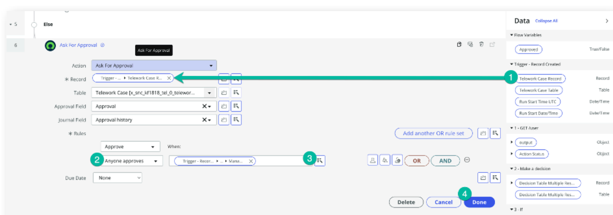
    1. Select "Action" as the step type.

    2. Select "ServiceNow Core" > "Ask for Approval".



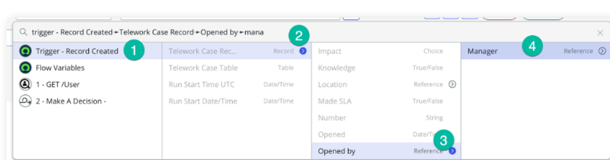
28. Setup the approval step.

1. Drag and drop the Telework Case Record from the data pills on the right-side of your screen.
2. Set the Approval When field to "Anyone Approves.
3. Dot-walk to the manager of the person that opened the telework request (see step 29 below).
4. Click the Done button.



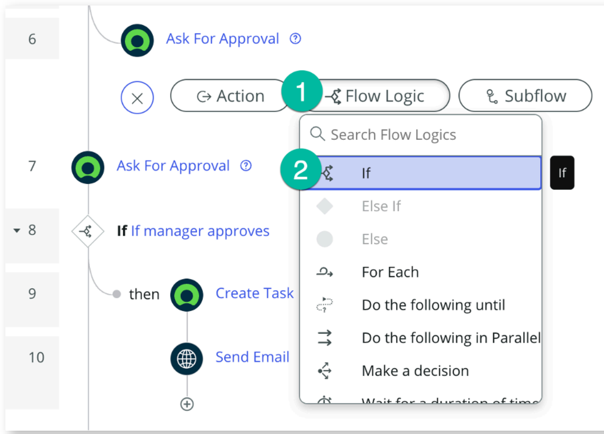

29. Dot-walk to the manager (step 3 above).

1. Click "Trigger - Record Created"
2. Click the blue arrow next to the "Telework Case Record" record to drill down.
3. Click the blue arrow next to the "Opened by" Reference to drill down.
4. Click on "Manager" to set the value.



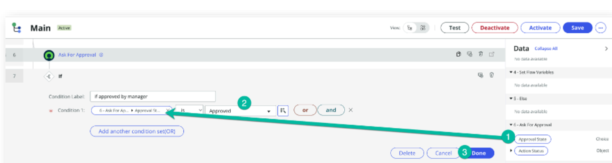
30. Add the if condition for then manager approves.

   1. Select "Flow Logic" as the step type.
   2. Select "If" from the list.



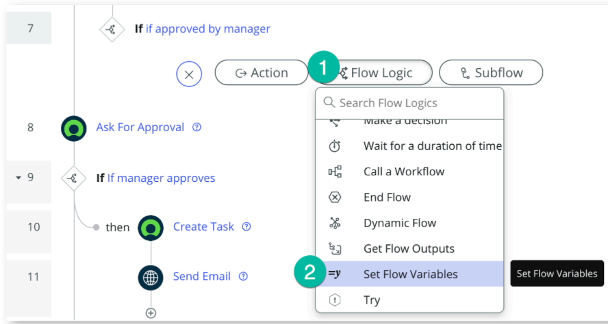

31. Setup the if condition to check for manager approval.

   1. Drag and drop Approval State from the data pills on the right-side of your screen.
   2. Set the value to "Approved".
   3. Click the Done button.

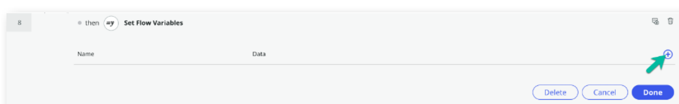


32. Set the Approval flow variable to true.

    1. Select "Flow Logic" as the action type.
    2. Select "Set Flow Variables" from the list.

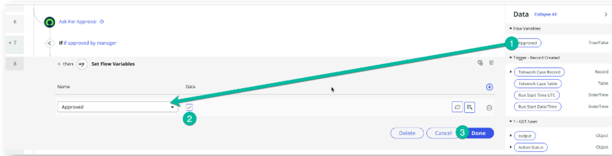



33. To add a flow variable to set, click the plus sign ⊕.



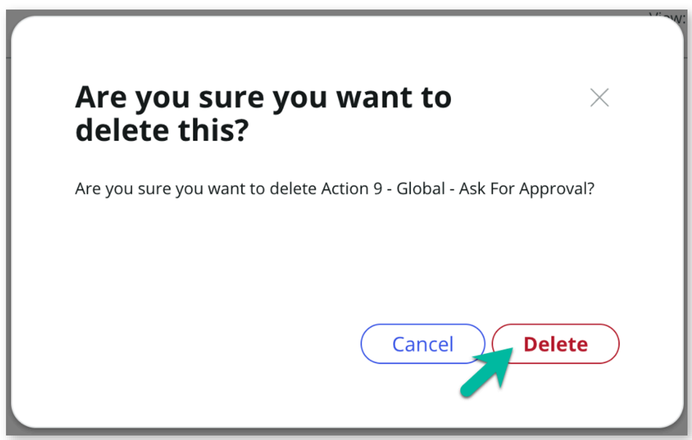

34. Set the Approved flow variable to true.

    1. Drag and drop the Approved flow variable from the data pills on the right side of the screen into the Name field.
    2. Check the data box for true.
    3. Click the Done button.



35. We no longer need our old Ask for Approval action, so go to line 9 and click the delete icon.


36. Confirm that we want to delete the action by clicking "Delete".

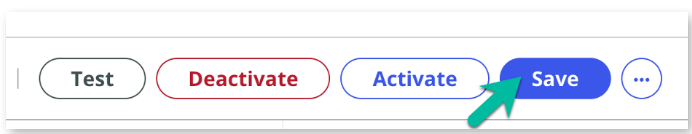



37. To modify the if condition to check for approval, click on line 9 to expand.

38. Modify the approval condition.

1. Change the condition label to: if approval conditions met
2. Clear Condition 1.
3. Drag and drop the Approved flow variable from the data pills on the right side of the screen into the Condition 1 field.
4. Set value to "True".
5. Click the Done button.

39. Save the changes to the Main flow by clicking the Save button.




40. Activate the Main flow by clicking the Activate button.

## Bonus

Do you remeber hot to test your flow from the previous lab? Test your work using what you've learned!

# Smarter Forms

## Purpose

Amandas team is growing and so is the backlog for what the requestors would like in the Telework Case management application. There are two things that need to be addressed:

For Amanda's team, they need to be able to allow anyone to on their team to work on a telework case but right now, not everyone knows what to do when they log in - what should they look for? How does the process progress? For this issue, Amanda would like to implement 'Playbooks' into the workspace to guide the fulfillers through the process.

The second backlog item they need to implement is having the requestor double check their manager. Right now, if there's no manager on the sys_user table, then the workflow stops and Amanda has to correct the issue. To mitigate this, Amanda would like to add a 'Manager' question for verification on the record producer and have it auto-populate with the manager on file for ease of use.

In this section of the lab, we'll be using Process Automation Designer, UI Builder, and Dev Studio.

# Notes

- 

# Workspace – adding a playbook to the record page view

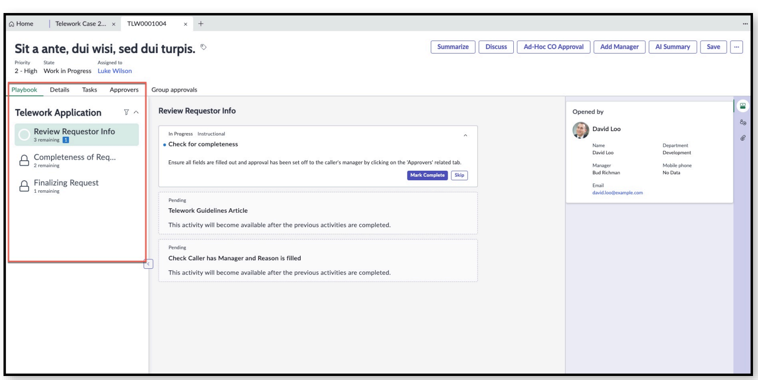
Amandas team is growing and her workspace that she uses to manage and report on telework cases is now used by more members of her direct reports. She has noticed some of them aren't always >aware of what to validate on the requests and Amanda would like a way for anyone who views the Telework Cases to be guided through what to look for and validate ensuring the process is followed >and all cases are treated equally.

The solution for this is to implement a playbook into the workspace.

Playbooks allow workspace users to interact with a business workflow in real time from within Workspace. Agents can use Playbook to update records, upload attachments, and complete tasks across multiple workflow activities.

Click Here for More Info

Example of a playbook in a workspace:



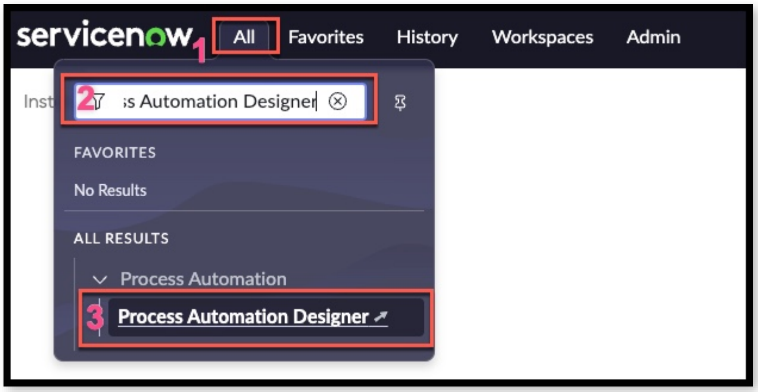
### Steps to Adding a Playbook

1. Design your playbook in Process Automation Designer
2. Test and Publish your playbook
3. Add playbook to your workspace in workspace builder
4. If you want to change the order of where the tab appears, use UI Builder

### Let's get started

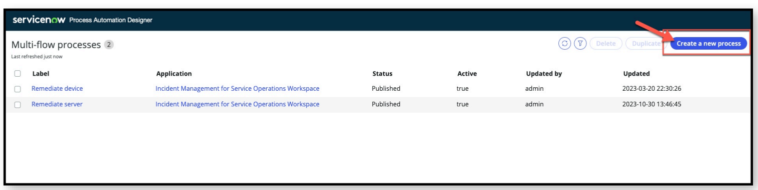
# Design your playbook in Process Automation Designer

1. Go to 'All'
2. Type in 'Process Automation Designer'
3. Click on "Process Automation Designer' (this will open up a new tab in your browser)



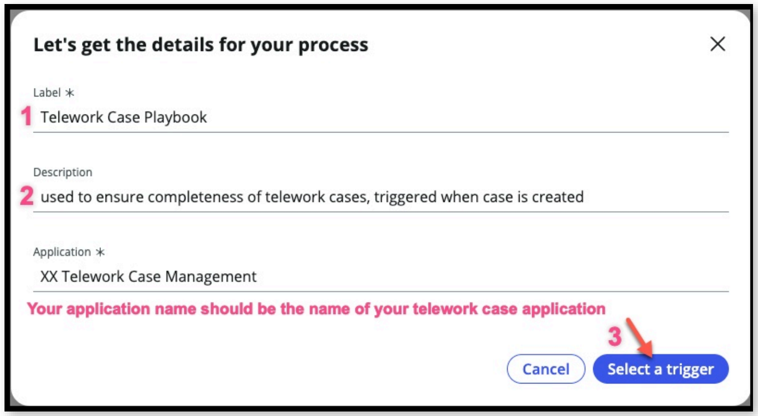

You are now in the Process Automation Designer list view showing you all the processes that have been completed.

> 1. Click on 'Create a new process'




Give your process a label and description (note: your application name should be the name of your Telework Case Management application, might be different than the one in the screenshot)

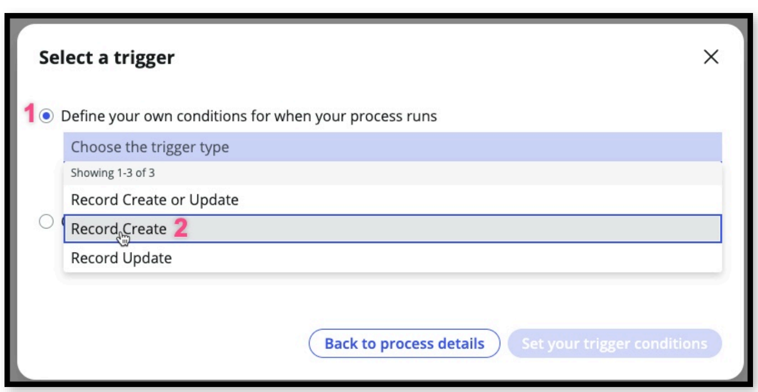
> 1. Label: Telework Case Playbook
>
> 2. Description: used to ensure completeness of telework cases, triggered when case is created
>
> 3. Click on "Select a trigger"

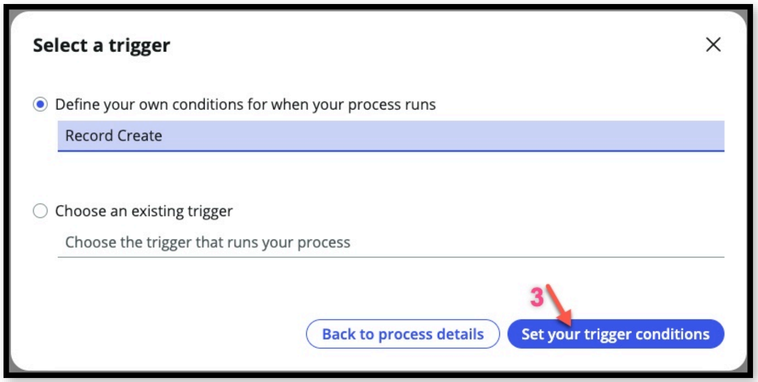



> Tip This is going to feel a lot like flow designer – in the background, flows are being triggered and used as we put together our process.

Select your trigger

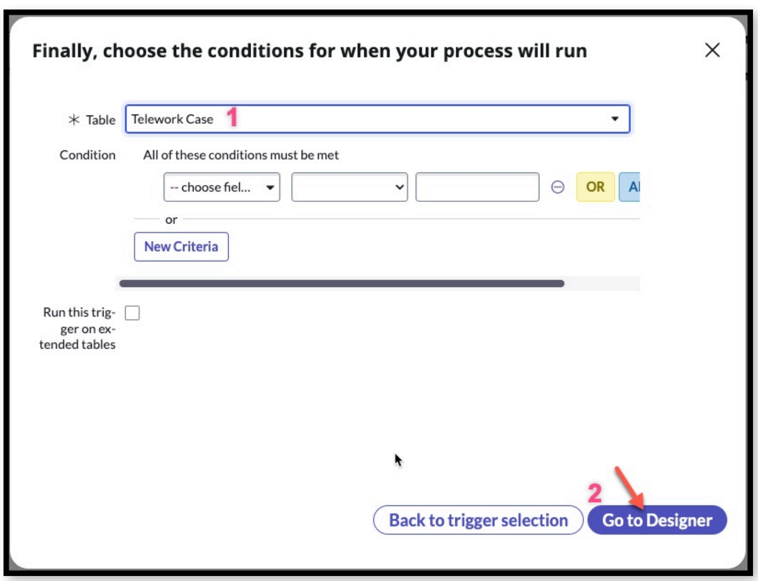
> 1. Choose the first option, "Define your own conditions for when your process runs"
>
> 2. Choose "Record Create"
>
> 3. Click on "Set your trigger conditions"





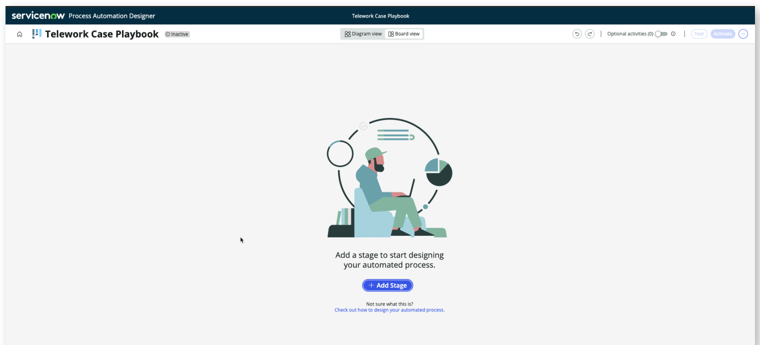

Finally, choose your table and conditions – we aren't going to have any conditions on types of records but that is an option.

> 1. Table: Telework Case
>
> 2. Click on "Go to Designer"




Switch to 'Board View'


You should now be on this page:




We're going to add 2 Stages:

> 1. Stage 1: Completeness of Request -This will be used to walk the workspace user reviewing the form and ensuring there's everything needed to make a good decision
>
> 2. Stage 2: Finalizing Request -This stage will be used to review that approvals are completed and IT has the task

## Go ahead and create the two lanes

> Click on: + Add Stage
>
> 1. Label: Completeness of Request
>
> 2. Description: review all information

3. When to start: When process starts

4. Click on 'Save and close'

Click on: + Add Stage (next to the stage you just created)

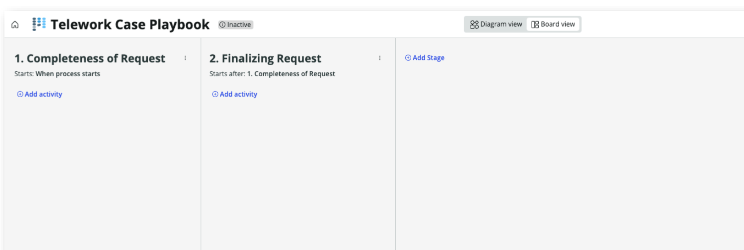
1. Label: Finalizing Request

2. Description: ensure all approvals and tasks have been created and completed

3. When to start: After specific stages

4. Click on 'Save and close'

Your screen should now look like this:



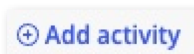

In the first Stage, Completeness of Request, we are going to have 3 activities:

1. Check the opened_by has a manager, the reason field is filled out and set the priority

2. See if the opened_by has any telework cases already submitted

3. Give the workspace user the telework guidelines knowledge article if they want to attach it

Completeness of Request – Activity 1

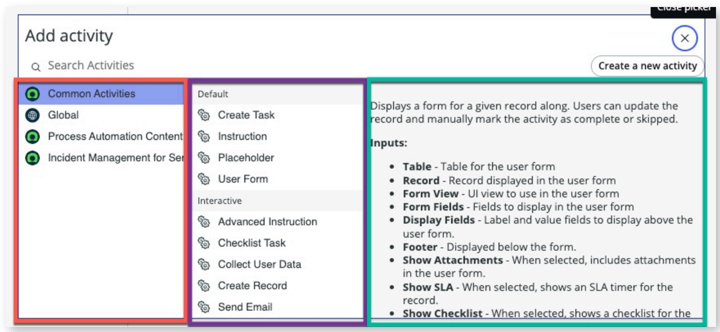
> 1. Click on +Add activity under Stage 1
>
> 
> 

## **Take A Break** Explore the Activity Screen

- 'Search Activities'
- 'Create a new activity' – to learn more about this, go here

*(the boxes have been outlined with color to differentiate them. In your instance, there will not be color around each section)*

- Red box: These are your bundled activities; in flow designer these would be your 'spokes'
- Purple box: In each Bundled activity, are the activities you can assign to your process. These can include interactive and non-interactive activities.
- Green box: For each activity, this box outlines what necessary info is needed for the activity to work.




More Info, Click Here

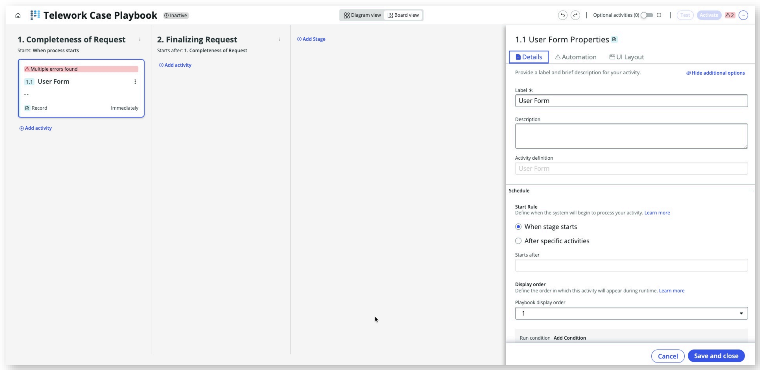
**Back to Lab**

We'll be using the 'Common Activities' bundle. Feel free to see what is under 'Global' and 'Process Automation Content' for ideas on what else you can add to a playbook.

Choose

> 1. Common Activities
>
> 2. User Form

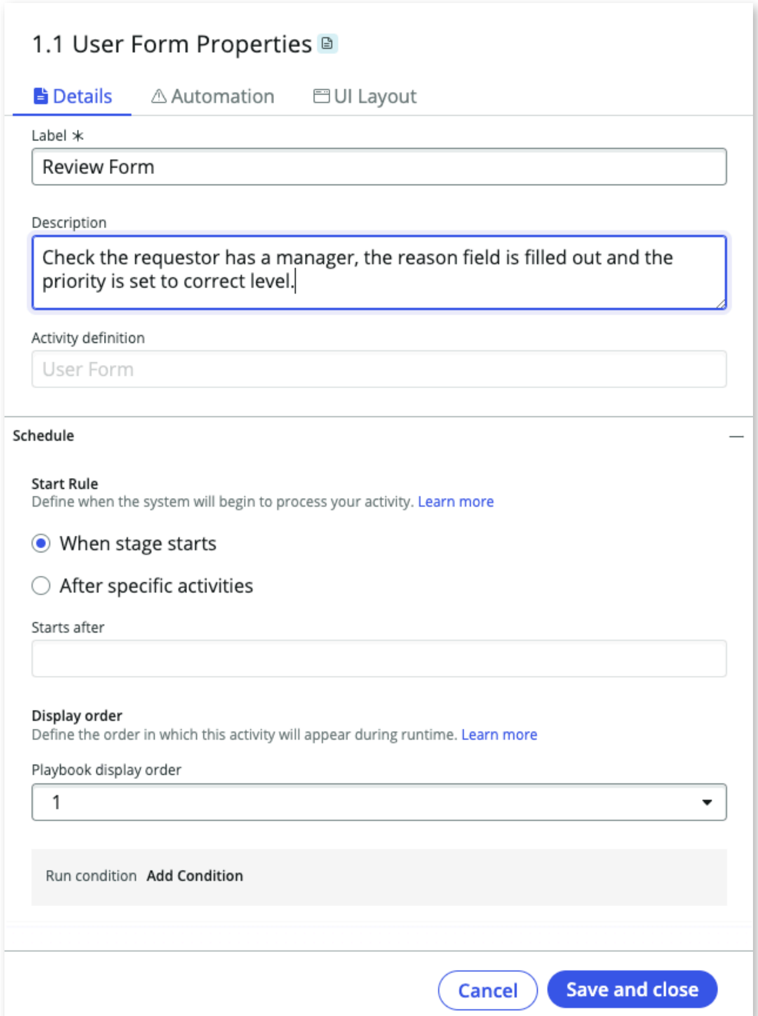
You should now be here:




On the right hand side, is the activity properties. This is where we give the necessary info for the activity to work.

Fill out the form (Edit activity properties):

> 1. Label: Review Form
>
> 2. Description: Check the requestor has a manager, the reason field is filled out and the priority is set to correct level.



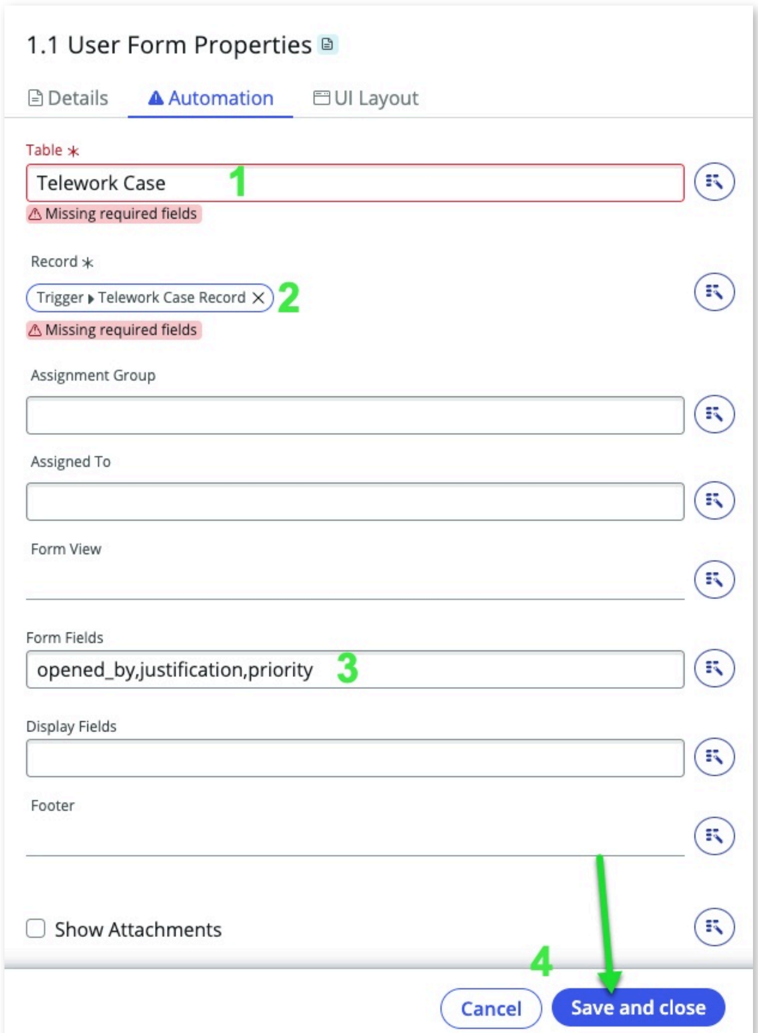

Click on 'Automation' tab (next to 'Details', DO NOT hit 'Save and close' yet, you will get an error)

This activity has some inputs it needs:

> 1. Table: Telework Case
>
> 2. Record: use the data pill picker to choose the trigger record, just like in flow designer.

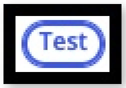
3. Form Fields: opened_by,justification,priority (this is a string field, must type them in) All other fields do not get changed.

4. Click on: 'Save'



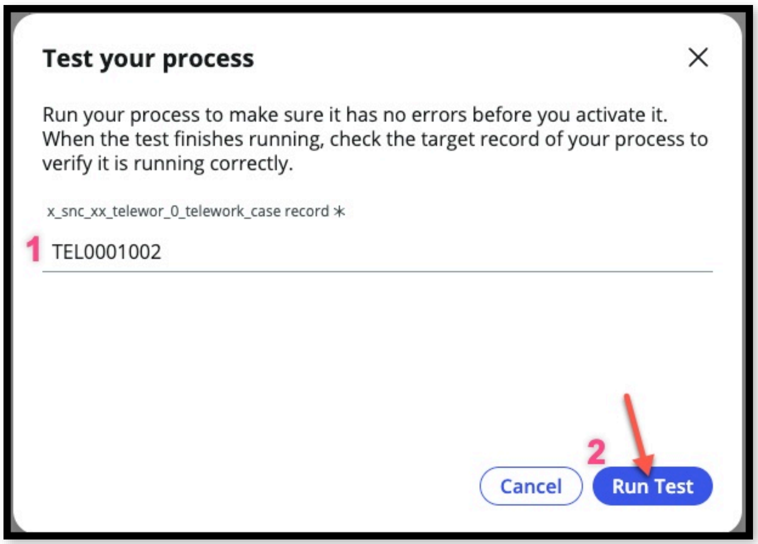

Before we go any further, let's test this out so you can see how this all will look to the workspace user.

Click on 'Test' in upper right corner near activate:

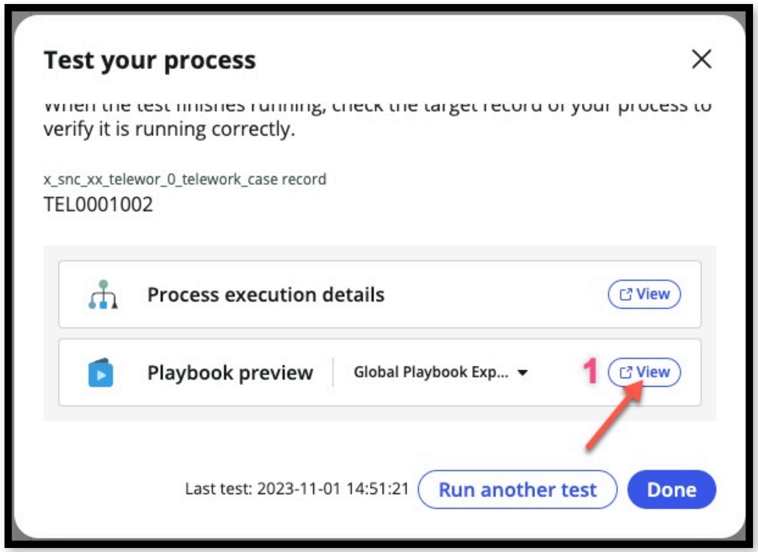



1. Choose any Telework Case record – the pictures below will be using TEL0001002

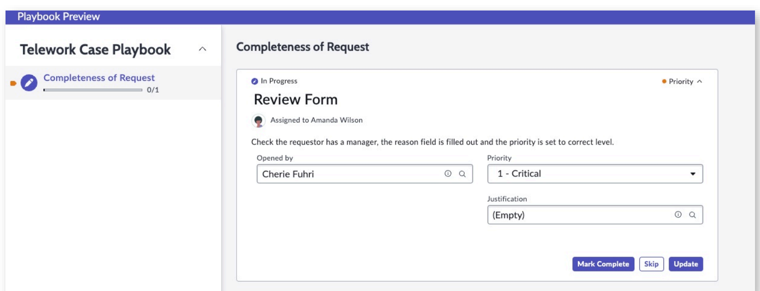
2. Click on: 'Run Test'




Choose the 'Playbook preview' – click on: View (it will open in a new browser tab)



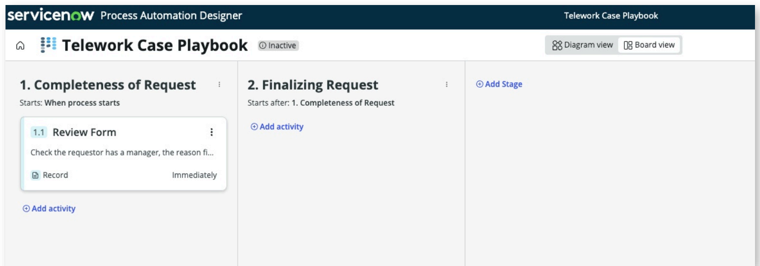

Give this screen a moment to load – when it's done, it should look like this:




Try adding something for 'Justification' then 'Update' then 'Mark Complete'

In the bottom right corner is a 'cog' – you can open this up and try some of the extra settings. These can be configured in the experience tab for your playbook (not doing in this lab).

If everything looks good, you can close this browser tab then click on 'Done' and you should be back to your playbook designer here:

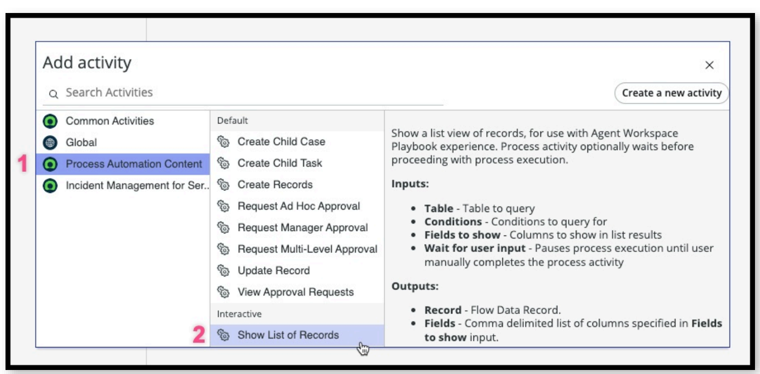



After the workspace user has reviewed the form, we would like them to see a list of any telework cases that have already been opened by the requestor. That will be our next activity – adding a list of telework cases opened by the requestor.

Underneath 'Review Form' activity, click on: +Add activity


Choose:

> 1. Process Automation Content
>
> 2. Show list of records

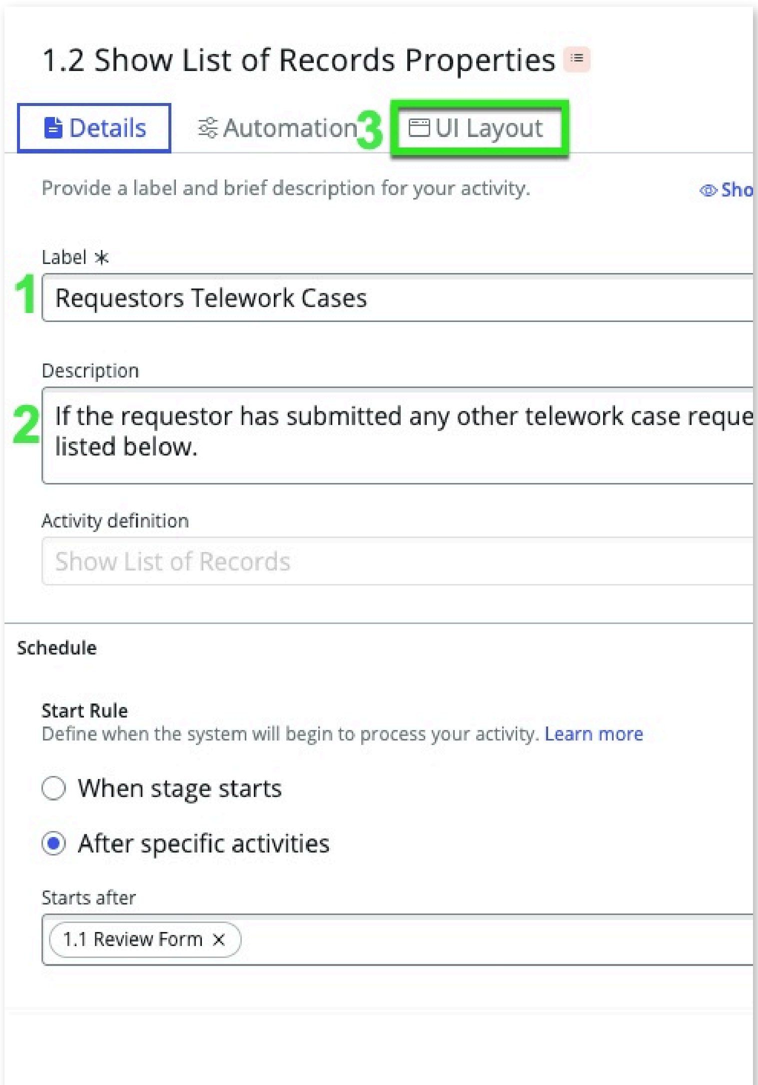



In the 'Edit activity properties' on the right hand side, fill out the form:

1. Label: Requestors Telework Cases

2. Description: If the requestor has submitted any other telework case requests, they will be listed below.

3. Click on 'UI Layout' tab




Fill out the 'UI Layout' tab

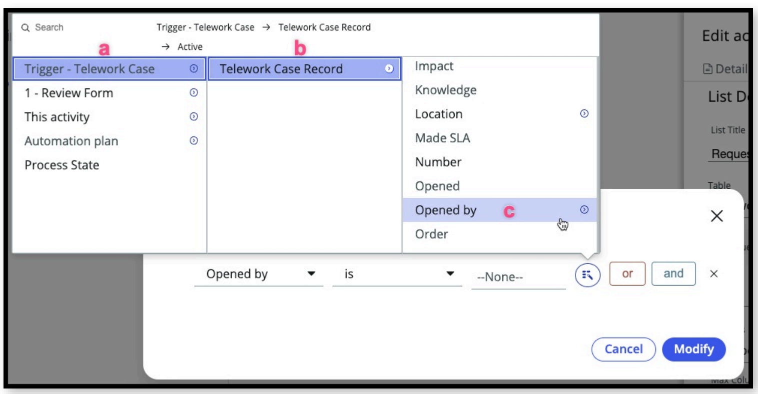
1. List Title: Requestor Cases

2. Table: Telework Case

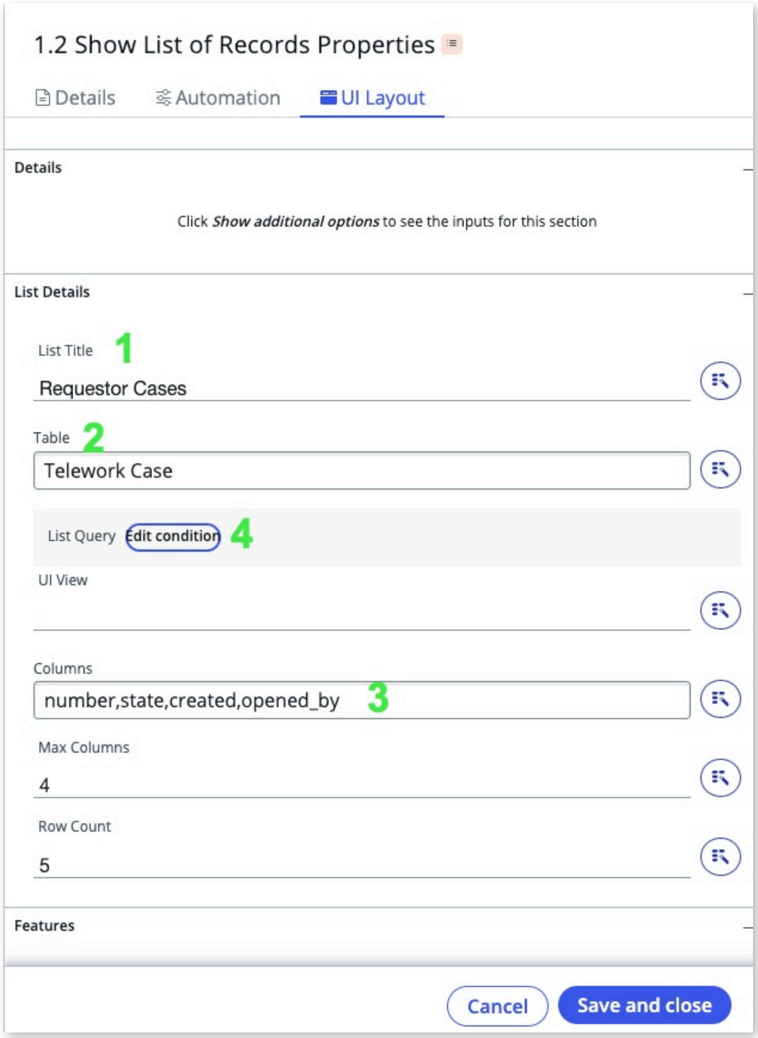
3. Columns: Number,state,created,opened_by

4. Click on 'Add Condition' next to 'List Query'
   > We need to only pull the records that have been opened by the current requestor.
   > Opened_by is opened_by (use the data pill picker)

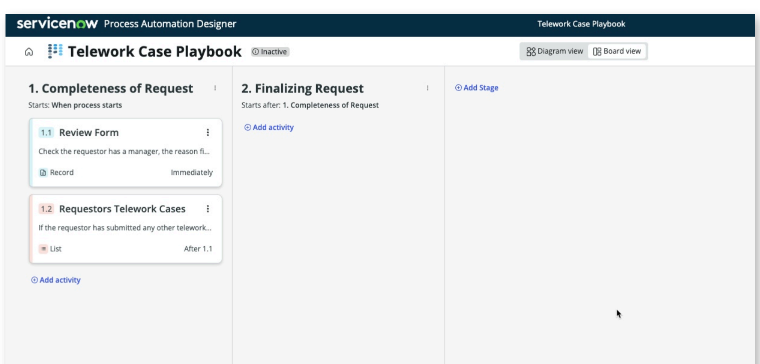
Step 4 'Add Condition' will look like this (then click 'modify'):




Before clicking on 'Save and close' your form will look like this:




Click 'Save and close' and now your process looks like this:

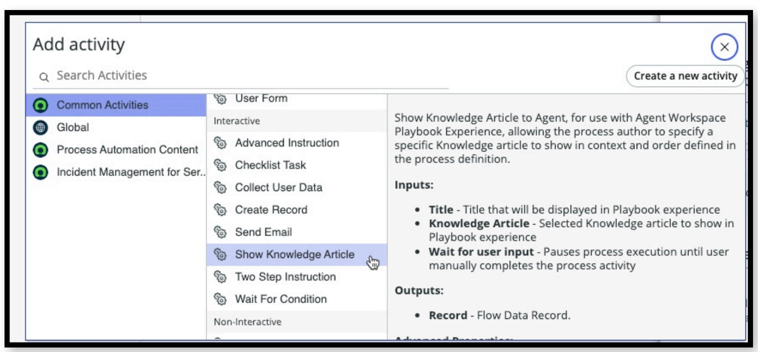


The last activity in this lane will be to show the workspace user the knowledge article on telework guidelines – many have asked to have this handy in case they need it at during the process or to attach for the requestor to review.

Click on +Add Activity under 'Requestors Telework Cases'


1. Common Activities

2. Show Knowledge Article



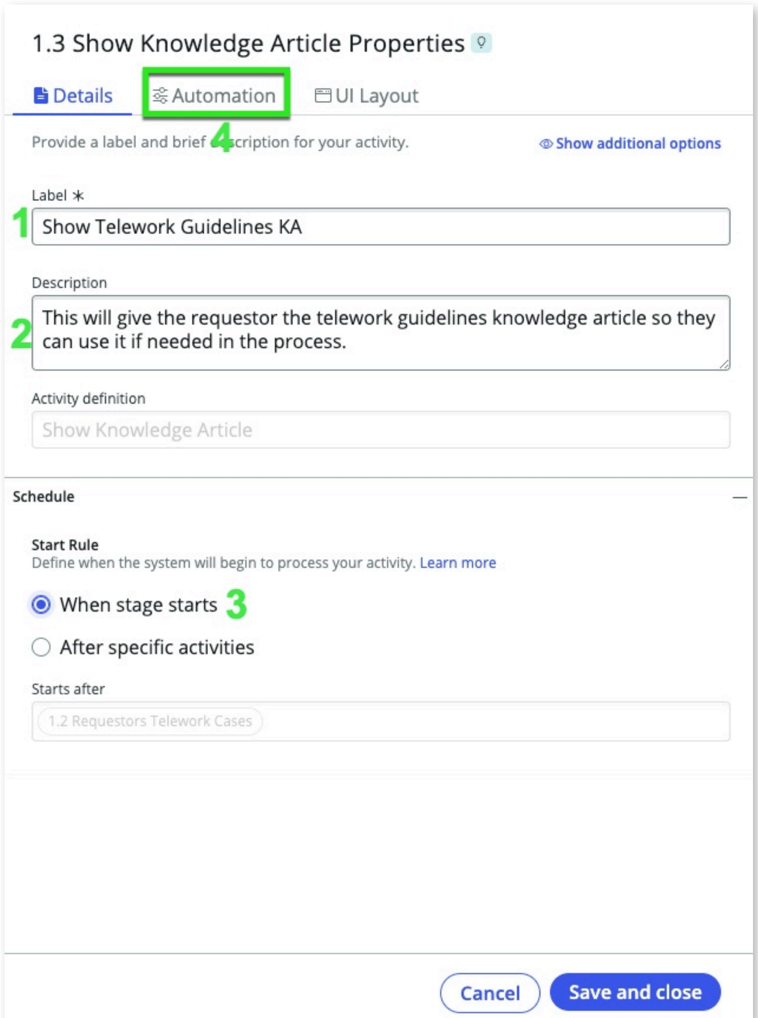

Fill out the 'Edit activity properties'

1. Label: Show Telework Guidelines KA

2. Description: This will give the requestor the telework guidelines knowledge article so they can use it if needed in the process.

3. Start rule: When stage starts (we want this to be available right away)

4. Click on: Automation tab (next to details tab)

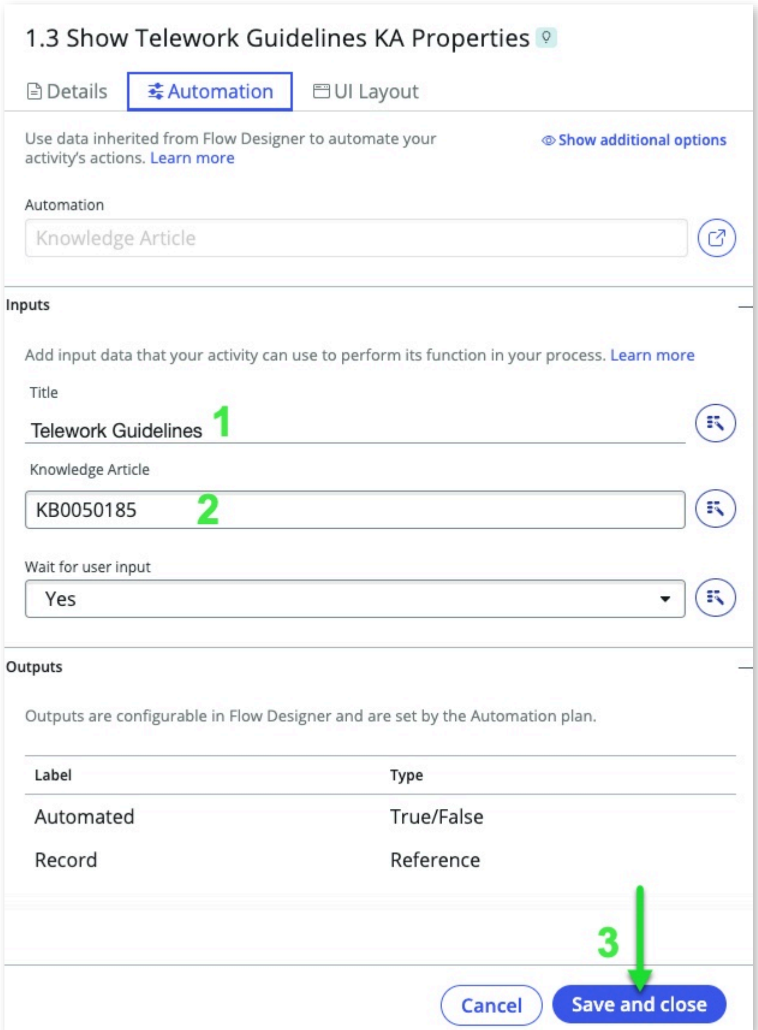



Fill out the 'Automation' tab

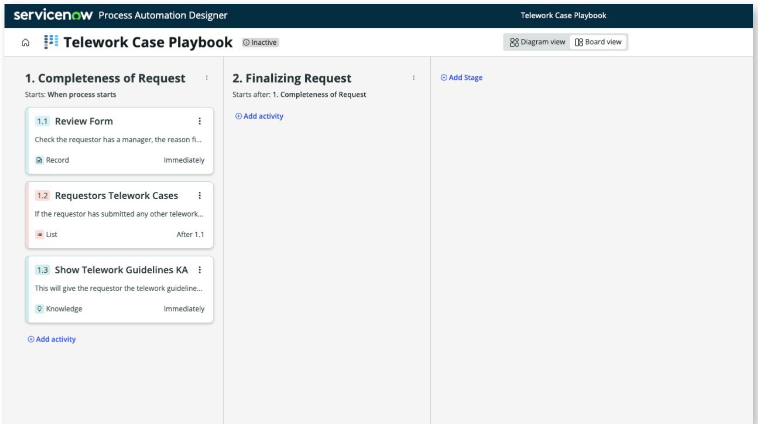
1. Title: Telework Guidelines

2. Knowledge Article: KB0050185

3. Click on: 'Save and close'

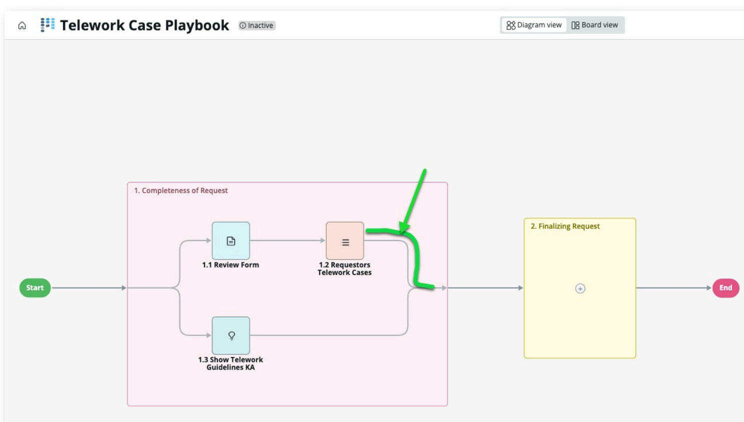



Super – your process should look like this now:




## If you have an error regarding an 'end connection' for step 2

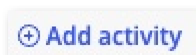
> Go to 'Diagram View'
> Connect Activity 1.2 to Stage 2
> It should look like this:




> then go back to Board view

# Feel free to take a break and test out your process like we did earlier

Now, let's add an activity to the 'Finalizing Request' stage and then we'll add the playbook to our workspace.

> Click on +Add activity in the Finalizing Request stage
>
> 
> 

Choose

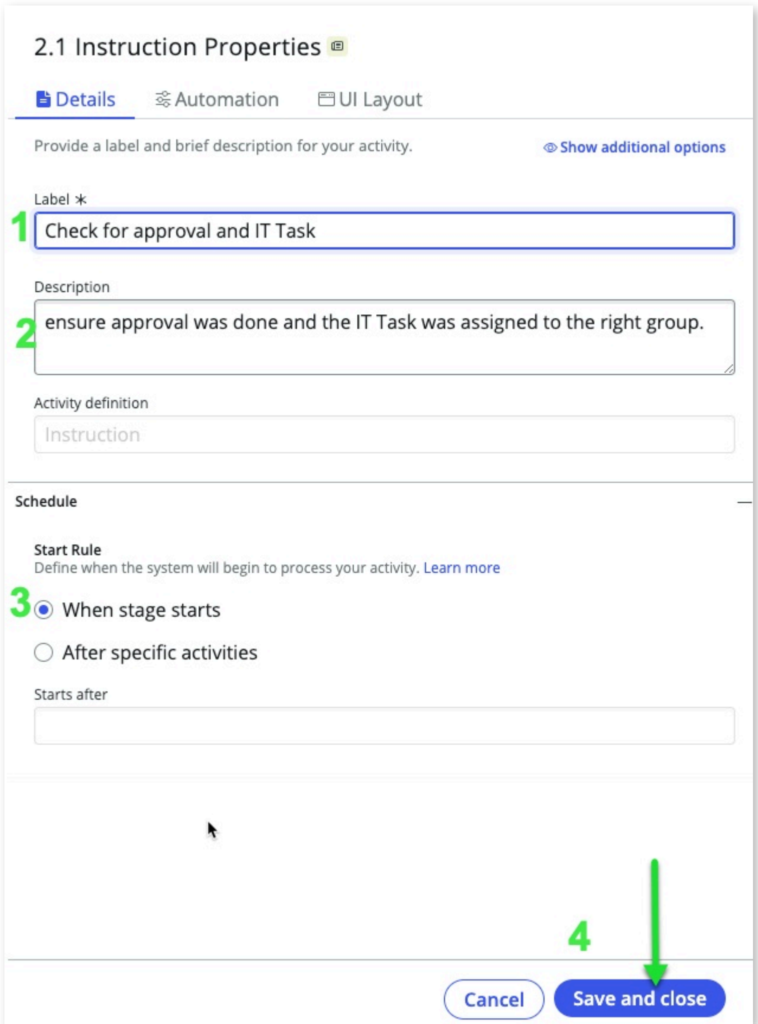
> 1. Common Activities
> 2. Instruction

Fill out the 'Edit activity properties' form

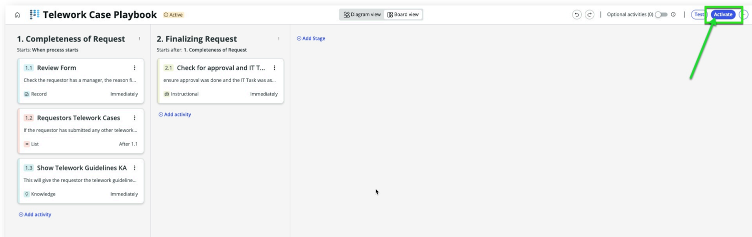
> 1. Label: Check for approval and IT Task
> 2. Description: ensure approval was done and the IT Task was assigned to the right group.
> 3. Start rule: When stage starts

4. Click on: 'Save and close'

That's it! If you look under the 'Automation' tab it automatically puts the description as your instruction. Your form should look like this:




Now, you're all set with version one of your Playbook! It's time to click on 'Activate'



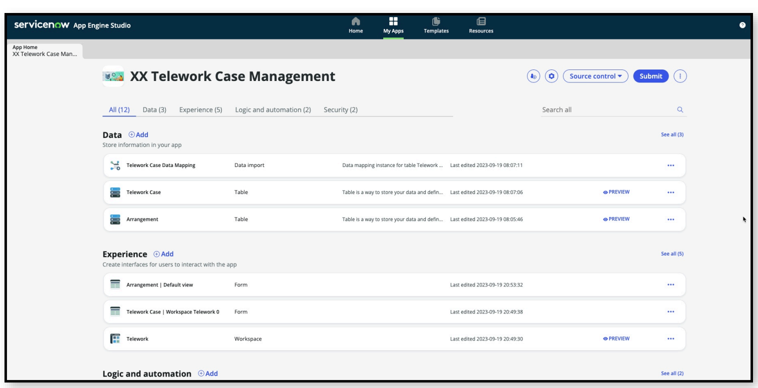

What else would you add to the process?

Did you notice an activity for manager approval? Some great activities already in there, out of the box!

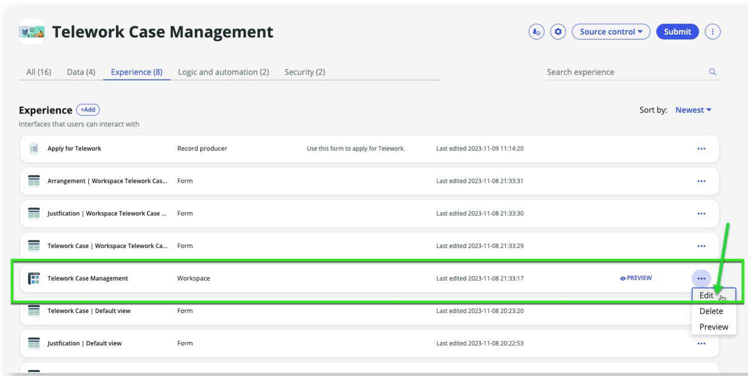
Now that we have published the playbook, we need to add it to our workspace in app engine studio.

> 1. Open 'App Engine Studio'
>
> 2. Open your 'Telework Case Management' application

You should now be here:

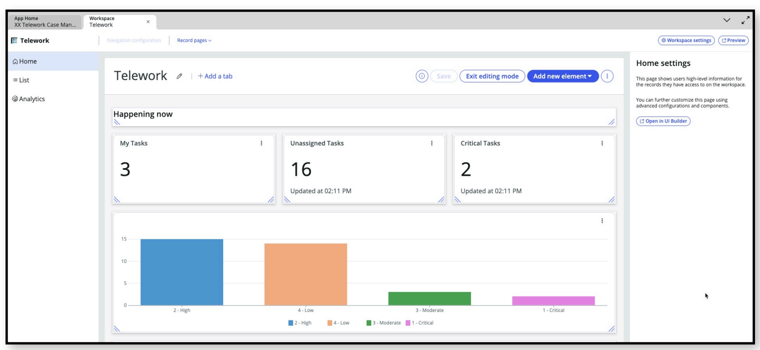



Let's edit the 'Telework' workspace




(if 'Get Started' pops up, go ahead and click on that)

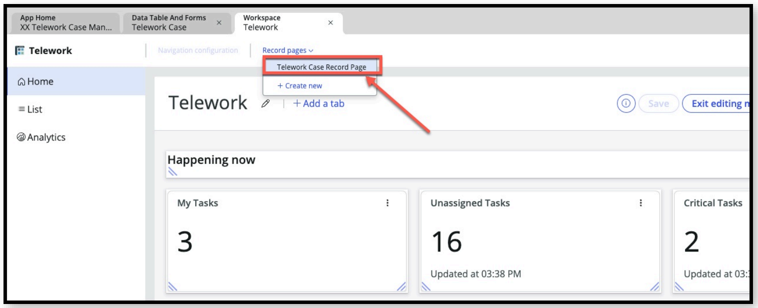
You should now be in the workspace builder, your screen should look something like this (some of the reports may differ)



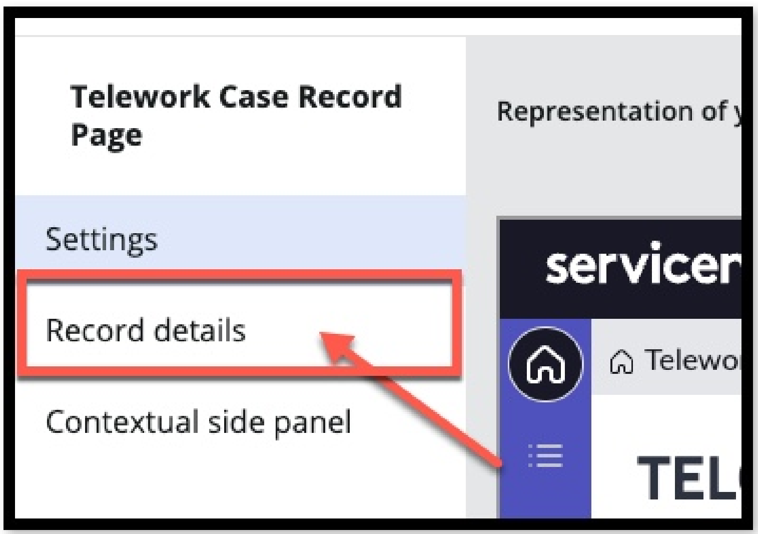

At the top of the page:

1. Click on 'Record pages'

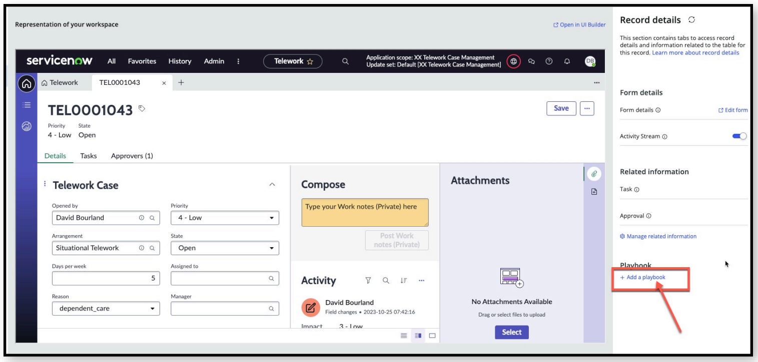
2. Change to 'Telework Case Record Page'




Choose 'Record details' on the left navigation bar:

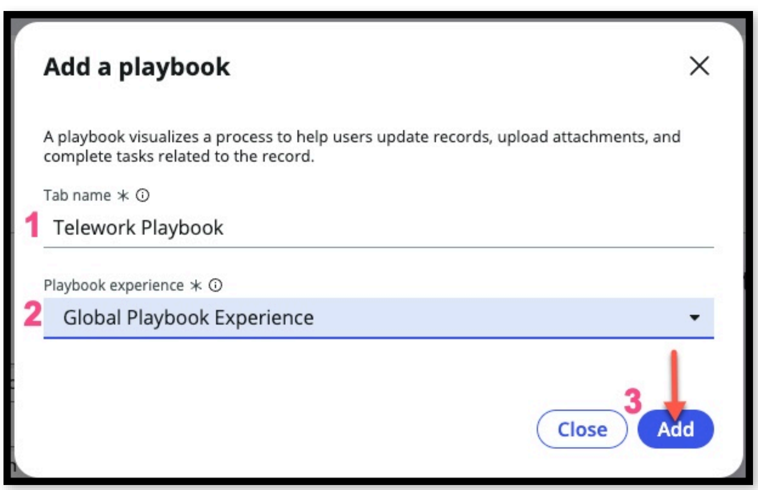



On the far right side, is the Record details configuration pane, choose + Add a playbook:




Enter in these fields:

> 1. Tab name: Telework Playbook
>
> 2. Playbook experience: Global Playbook Experience
>
> 3. Click on 'Add'




So, what is a playbook experience?

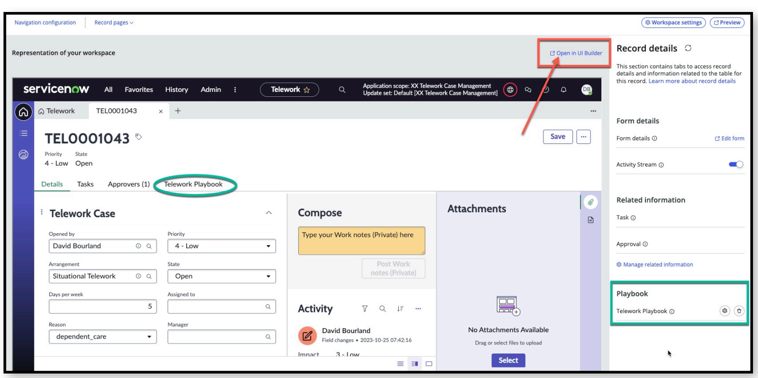
After creating your process in the process automation designer, you can either use the out of the box experience "Global Playbook Experience" which is great if you used all out of the box activities and like the way it looks in the workspace.

The other option is you can create your own experiences and components which can give you even more flexibility for the user to work. You can even use UI Builder to create new components and further create the exact experience your workers need.

For more on using playbook experiences, there are a lot of resources – here are a couple:

- Playbook Experiences – docs page
- Platform Academy Session #21 Video

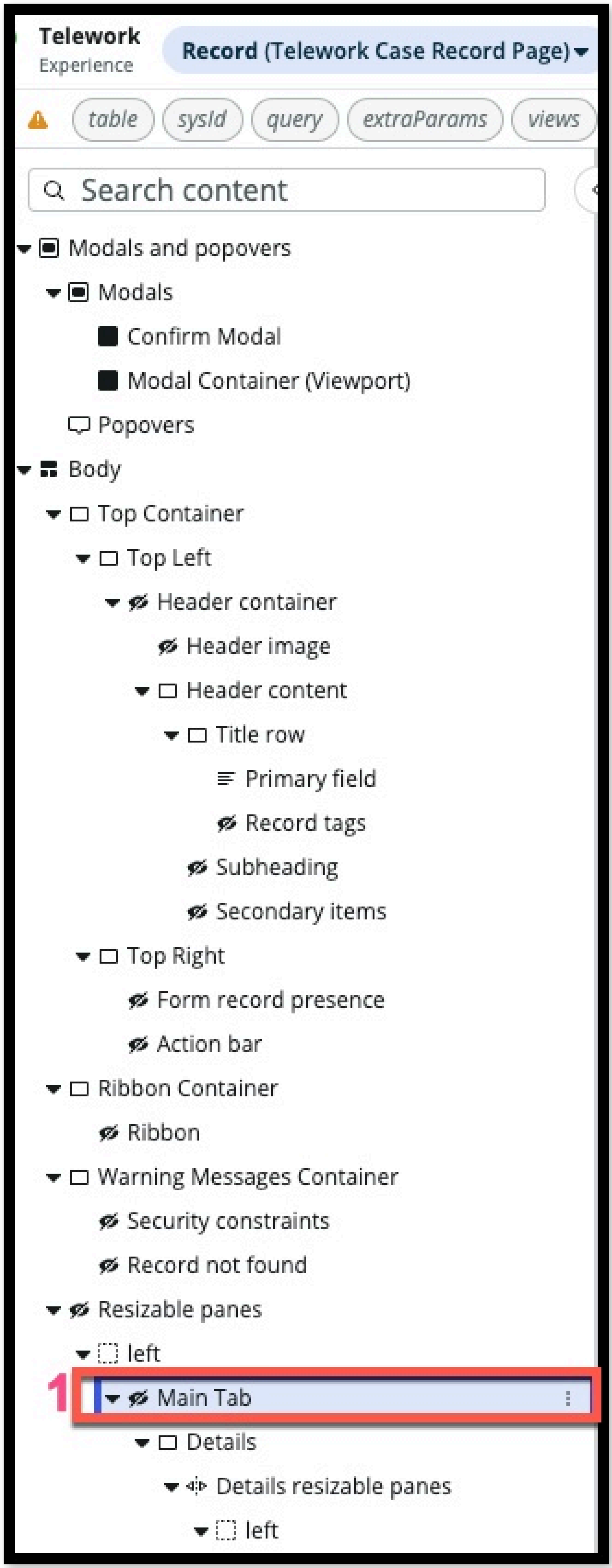
Now that you have added the playbook to the workspace, take note of how it's showing up as a tab on the record page at the very end, the last tab.
Let's move it to the very beginning. This is how it looks now – go ahead and click on the link in the upper right 'open in UI Builder' (and if there's a pop up, continue to UI Builder)




Now that you're in UI Builder – it takes a minute to load. Soon, you will see the pages and sections on the left.

> 1. On the left go to: (these should already be expanded)

> Body
>
> > - Resizable panes
> >
> > > - Left
> > >
> > > > - Click on "Main Tab"

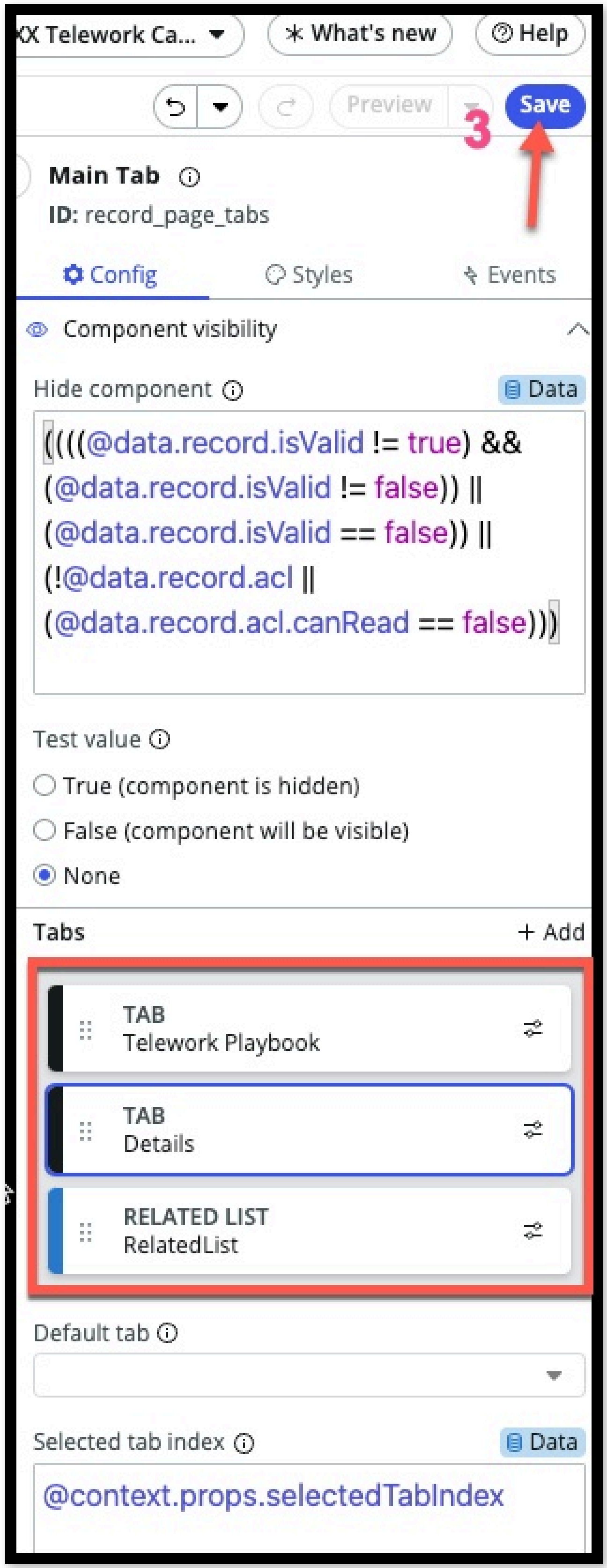

Telework
Experience
Record (Telework Case Record Page)
table
sysId
query
extraParams
views
Search content
Modals and popovers
Modals
Confirm Modal
Modal Container (Viewport)
Popovers
Body
Top Container
Top Left
Header container
Header image
Header content
Title row
Primary field
Record tags
Subheading
Secondary items
Top Right
Form record presence
Action bar
Ribbon Container
Ribbon
Warning Messages Container
Security constraints
Record not found
Resizable panes
left
1
Main Tab
Details
Details resizable panes
left

Now, on the right side pane of UI Builder

> Drag the Telework Playbook tab so it's on top of the Details tab.

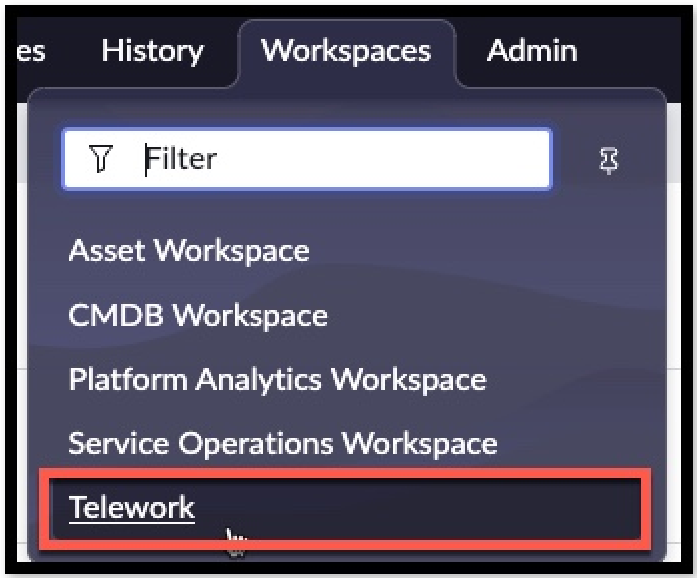
XX Telework Ca...
What's new
Help
Preview
Save
Main Tab
ID: record_page_tabs
Config
Styles
Events
Component visibility
Hide component
Data
((((@data.record.isValid != true) && (@data.record.isValid != false)) || (@data.record.isValid == false)) || (!@data.record.acl || (@data.record.acl.canRead == false)))
Test value
True (component is hidden)
False (component will be visible)
None
Tabs
+ Add
TAB
Details
RELATED LIST
RelatedList
TAB
Telework Playbook
Default tab
Selected tab index
Data
@context.props.selectedTabIndex

It should look like this and then Click 'Save':


XX Telework Ca...
What's new
Help
Preview
Save
3
Main Tab
ID: record_page_tabs
Config
Styles
Events
Component visibility
Hide component
Data
((((@data.record.isValid != true) &&
(@data.record.isValid != false)) ||
(@data.record.isValid == false)) ||
(!@data.record.acl ||
(@data.record.acl.canRead == false)))
Test value
True (component is hidden)
False (component will be visible)
None
Tabs
+ Add
TAB
Telework Playbook
TAB
Details
RELATED LIST
RelatedList
Default tab
Selected tab index
Data
@context.props.selectedTabIndex

You're done! When the 'Save' button in the upper right is greyed out, you're good to close this tab and now we'll test in your workspace.

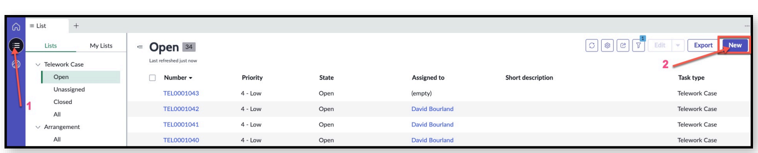
Go back to the platform view and under workspaces, find 'Telework':




The playbook runs when a new Telework case is created, so we'll need to do that in order to see our playbook in action.

Once the workspace opens:

> 1. Choose the 'List' action on the far left
>
> 2. Choose 'New'




Fill in the fields:

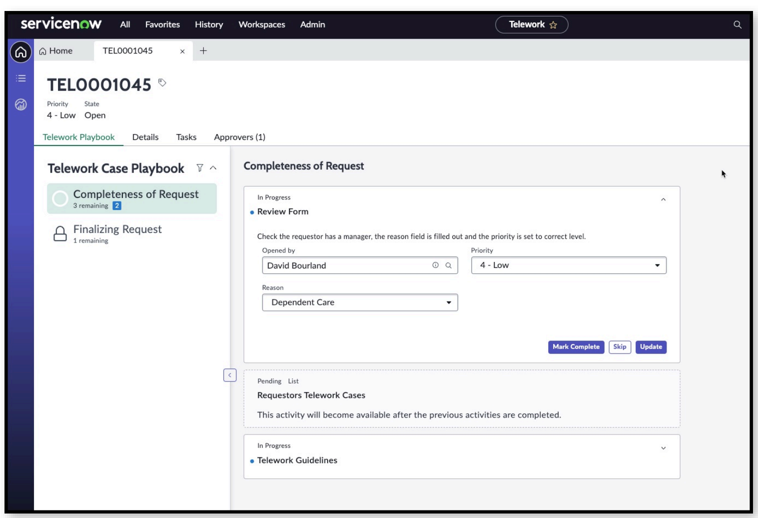
1. Arrangement: Remote
2. Justification: Dependent Care
3. Click 'Save'

(if it doesn't show up right away, refresh browser – this might happen the first time you run it)

Here's your playbook

Walk through it and check out the experience –
What would you do differently?
What else would be helpful?

Congrats!



# Record Producer

Verify Manager and Auto-populate 'Requested For' field

One of the complaints Amanda has had is sometimes the requestors don't have a manager listed or if there is one listed, it's the wrong one. This results in a delay of approval and ultimately a poor experience for the requestor.

The solution we've decided on is to list the requestors manager on the record producer, so they can either leave it or change it or fill it in.

This will require a new field on the telework table, changing some of the record producer questions, a script to auto populate a 'requested for' field, and using the 'Auto Populate' section on the record producer. Sounds like a lot, but it will go quickly.

Here are our steps:

1. Create a new field on the telework case table called "Manager" and "Requested For"

2. Create a new question on the record producer for "Manager" and "Requested For"

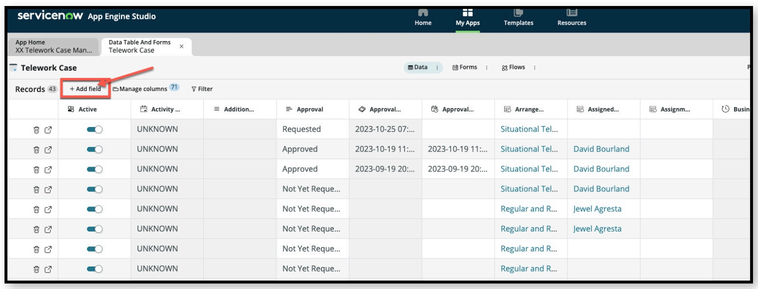
3. Create a script to auto-populate the 'Requested For' with the logged in user name using design studio

4. Change the flow to work with these new fields

## Creating new fields on the telework case table

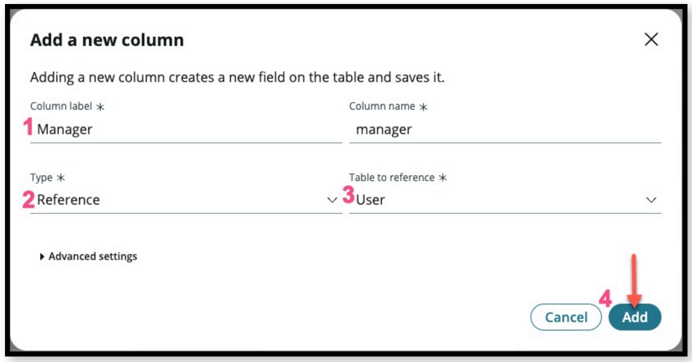
In app engine studio in the Telework Case application:

> Open the 'Telework Case' table (not shown)
> Click on: + Add field



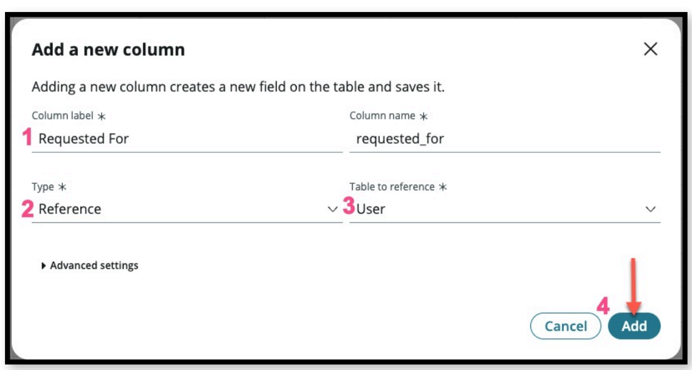

Fill in these fields:

1. Column label: Manager
2. Type: Reference
3. Table to reference: User (sys_user)
4. Click on: 'Add'



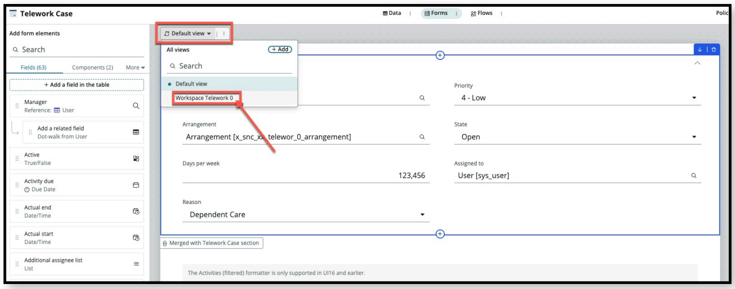

Now, add a new field 'Requested For' using the same steps and these attributes:

1. Column label: Requested For
2. Type: Reference
3. Table to reference: User
4. Click on: 'Add'



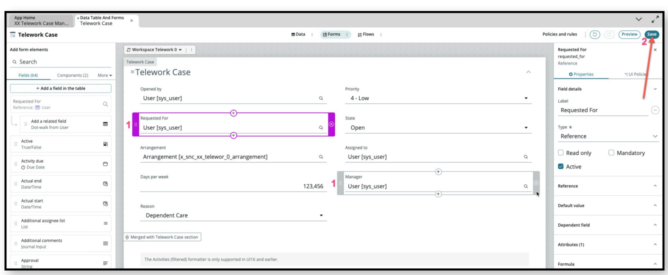
Let's add it to the workspace form

> Click on 'Forms' (center at the top, to the right of 'Data')
> Change 'Default view' to 'Workspace Telework'



1. Add the field 'Manager' and 'Requested For' to the form

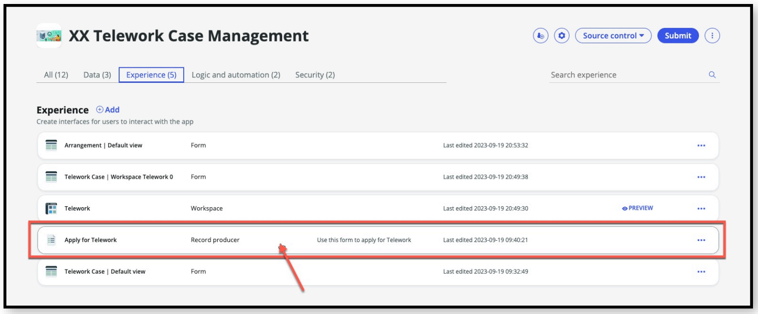
2. Click on 'Save'



Great – you now have a new field called 'Manager' and 'Requested For' and they're on your workspace form.

## Next, we'll add the question to our record producer

Go back to your App Home for Telework Case Management and open the Experience, 'Apply for Telework – record producer':




Once it opens up, choose:

> 1. Questions
>
> 2. Delete the Question Set – 'Standard Employee Questions'
>
> 
> 
>
> 3. After clicking on the 'x' click on 'Remove'

Next, we'll add in the question to find out who they are requesting the telework arrangement for – this will give the teams flexibility for people to request on behalf of someone else which has also been something the teams have been requesting be added to the form.

Click on 'Insert New Question':

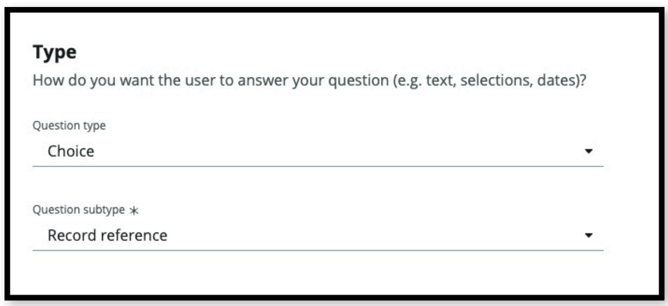
1. Question Type: Choice

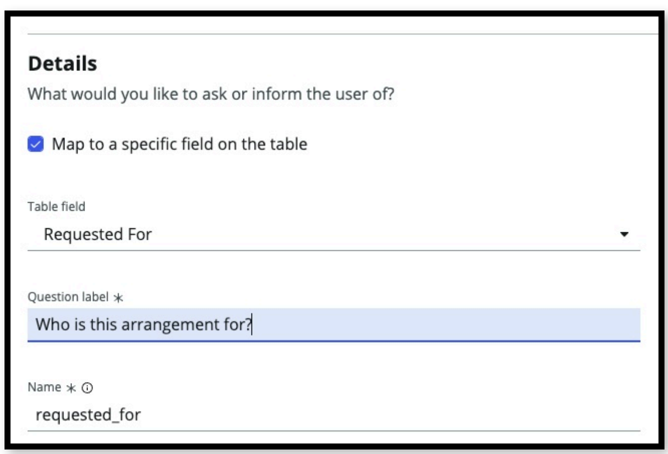
2. Question subtype: Reference

3. Map to a specific field on the table – True

4. Table field: Requested For

5. Question label: Who is this arrangement for?

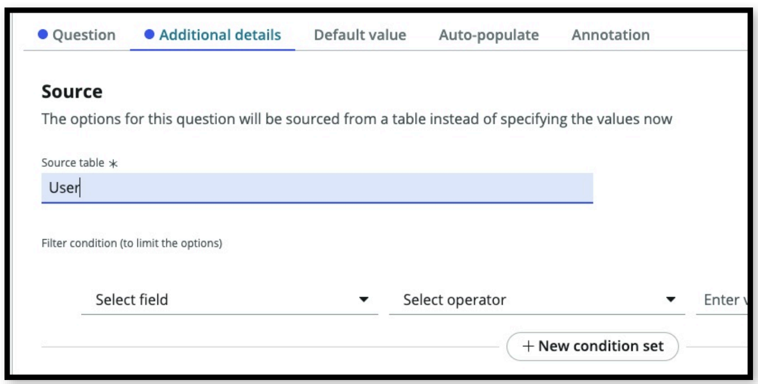


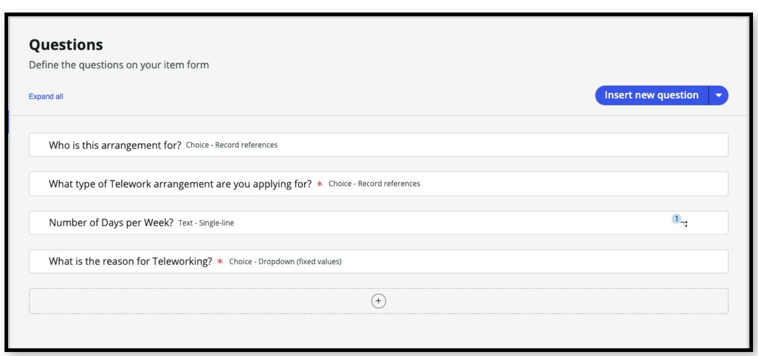

Click on 'Continue to Additional details'

1. Source table: User (sys_user) *no conditions

2. Click on 'Insert question' in bottom right


Move this question to the top of form, it should now look like this:

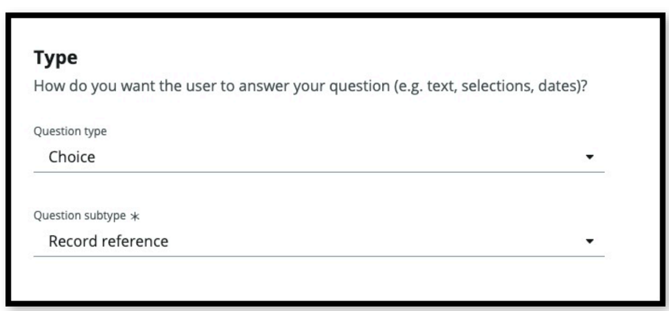


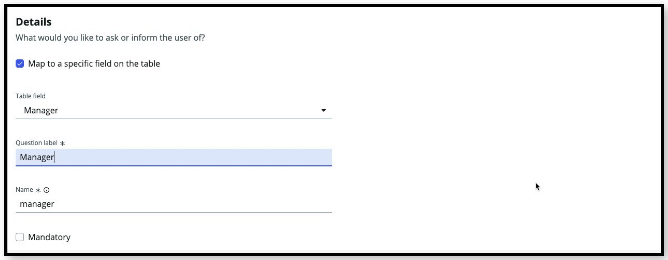
## Next we need to confirm we have the right manager

We'll add the manager question to this form and populate it based off of who they have in the 'requested for' question.

Click on: Insert new question


1. Question type: Choice
2. Question subtype: Record reference
3. Map to a specific field on the table: True
4. Table field: Manager
5. Question label: Manager

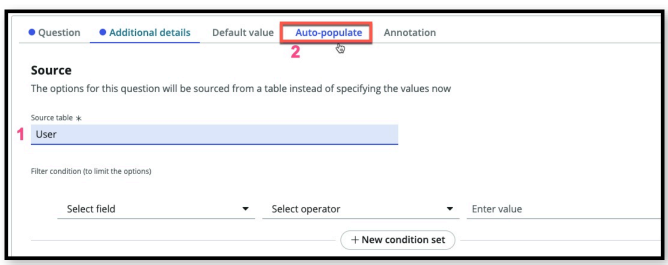





Continue to 'Additional details'

1. Source table: User (sys_user)

2. Click on 'Auto-populate'

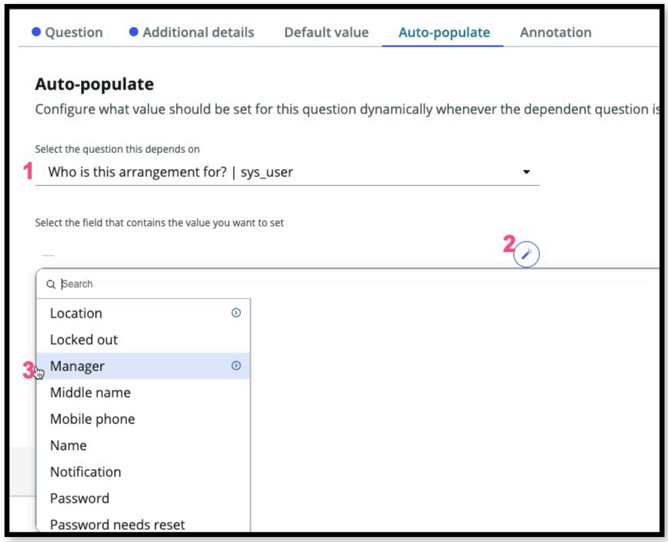



Fill out the form:

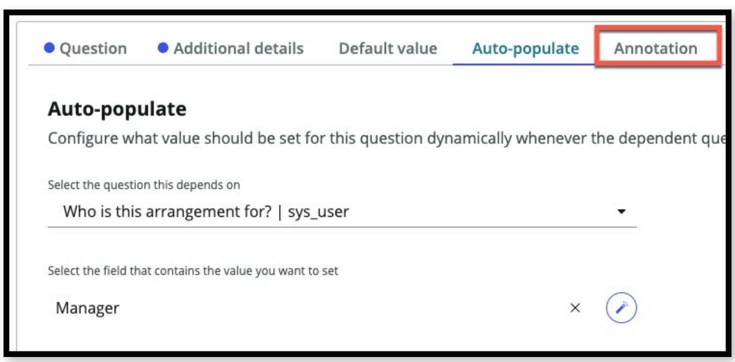
1. Select the question this depends on: 'Who is this arrangement for? | sys_user

2. Use the data pill picker button on the right

3. Select the field that contains the value you want to set: scroll to 'Manager'




Click on Annotation

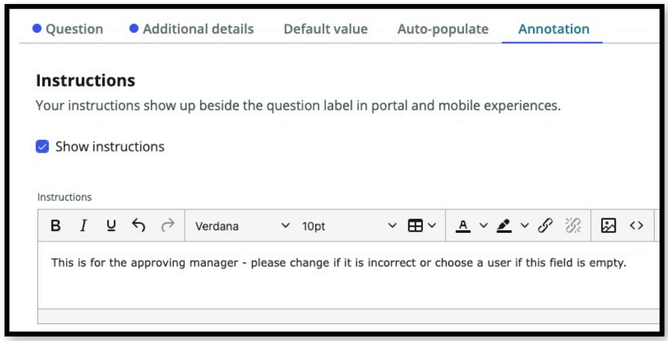



Fill out the form:

1. Show instructions: true

2. Instructions: If this is the correct manager don't change. If it is empty or the

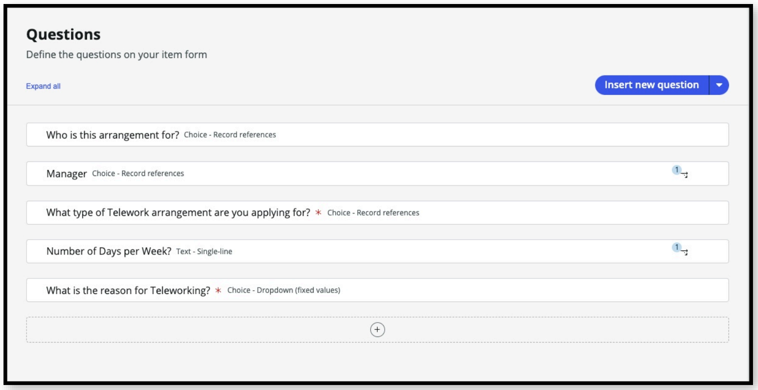
wrong manager, please change. The approval for your telework request will go to this person.




Insert question:


**Bonus** on your own, add a behavior to the 'Manager' question so it's mandatory if it's empty!

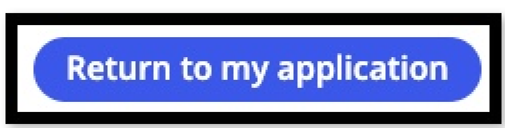
Move the 'Manager' question so it's the 2nd question, should look like this now:



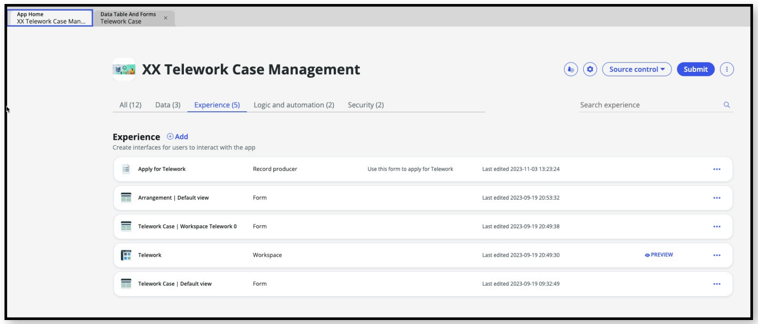

Click on 'Review and submit'


Click on 'Submit'


Click on 'Return to my application'




And you're here:




Next, we want to auto-populate the 'Requested For' question with whomever is filling out the form. To do that, we need to go into design studio, find the variable and add it to the default tab with one line of code.

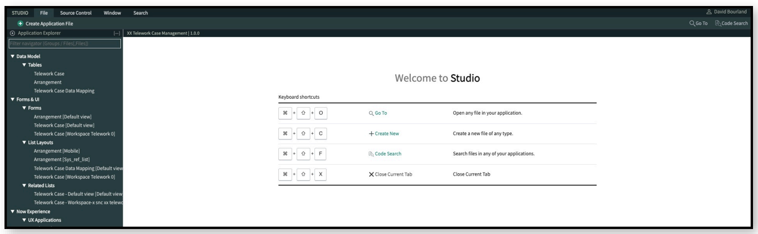
We need to do this part in the Dev Studio
To get to the Dev Studio:

1. Click on the 'cog' in the upper right


2. Click on 'open app in Dev Studio'

> 
> 

You should be here:



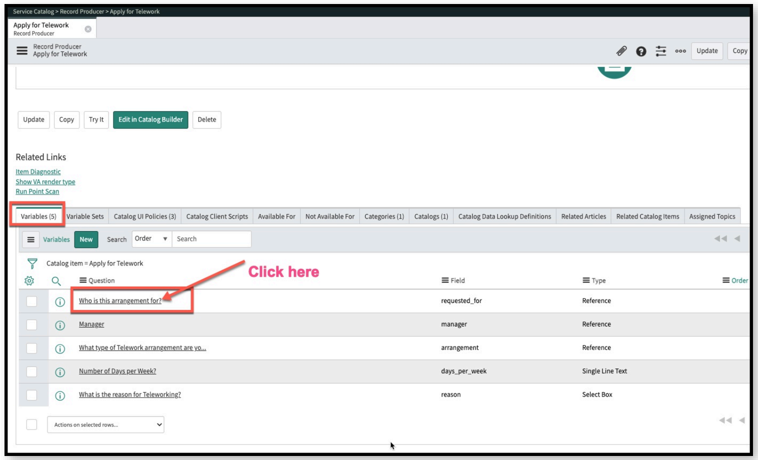

On the left hand navigation are all the application files we've created so far in our Telework Case application.

Scroll on the left to find

> Service Catalog
>> Record Producer
>>> Apply for Telework – click on 'apply for telework' record producer
>>>
>>> 
>>> 

This is your record producer platform default view.

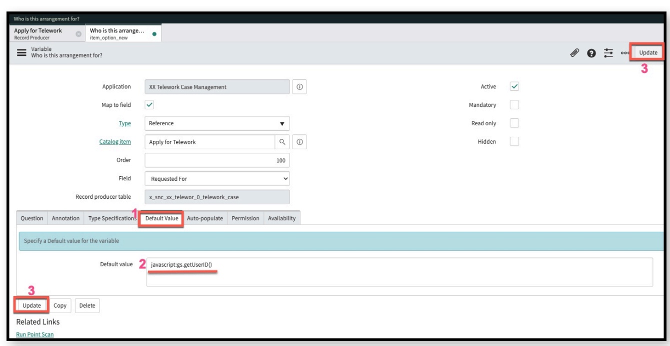
Scroll down to the related lists, 'Variables'.
Click on the variable 'who is this arrangement for?'




1. Scroll down to the tabbed section 'Default Value'

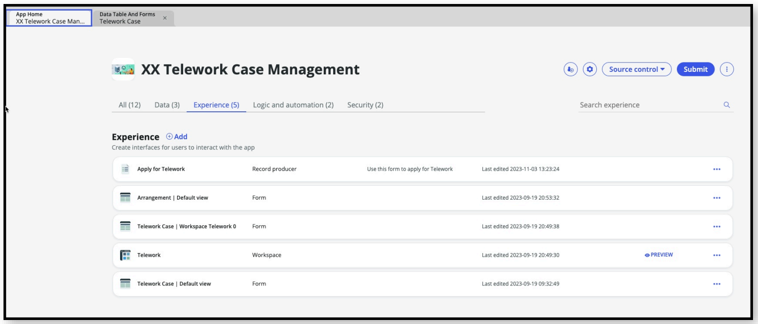
2. Enter into the box: javascript:gs.getUserID()

3. Click on: 'Update' (use either one, there's 2 update buttons)



After updating, it stays on this tab.

We're all done with the Design studio, you can close this browser tab and go back to the your Telework App home.



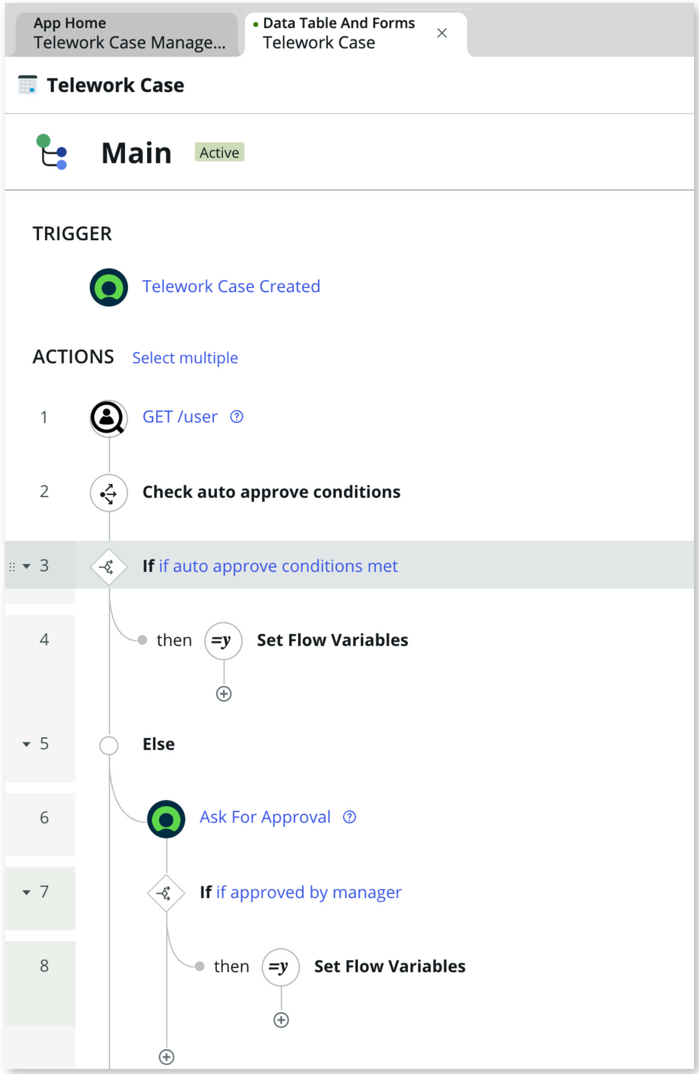

At this point

> > we've updated the data table modified the record producer experience

All that's left to do now is update the flow so it takes advantage of our new fields. Remember, we're relying on the 'requested for' field now, not the 'opened by' since sometimes they could be different.

### Updating Flow 'Main'

> - Click on 'Logic and automation'
> - Open the 'Main' flow (switch to tree view)




2 things we need to change:

1. Ask For Approval – change this to the 'manager' field

2. Send Email – send the email to the email of the 'requested for'

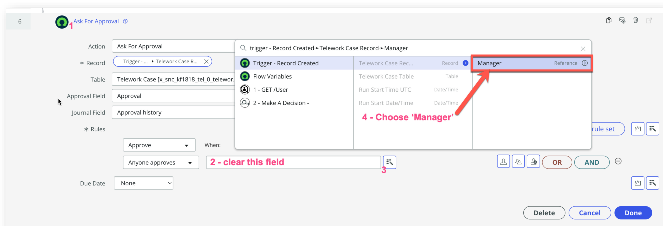
Changing Approval

1. Expand Step 6 – Ask for Approval

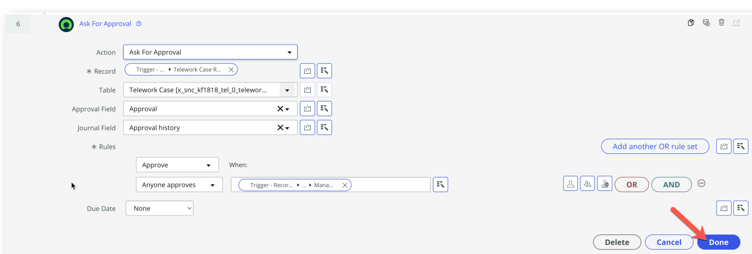
2. Clear the box after 'Anyone Approves'

3. Click on the data pill picker to find the Manager field

4. Trigger Record Created: tiny arrow: scroll and click on 'Manager' field




Click on 'Done'



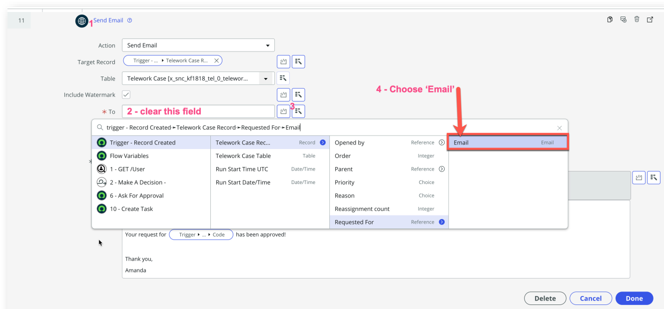

Let's change the email To field to the requested for email address

1. Expand 'Send Email' (step 11)

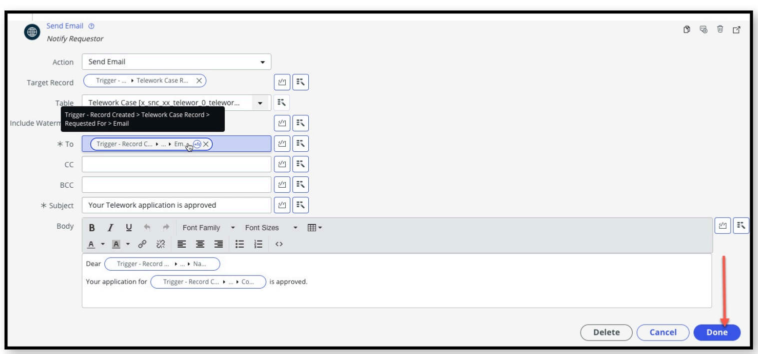
2. Clear the 'To' field

3. Choose the data pill picker for the 'To' field

4. Trigger – Record Created: Telework Case Record tiny arrow: Requested For tiny arrow: scroll and click on 'Email'




Click on 'Done'




Are there any other fields you need to change? Sometimes people use the opened_by name in the email body, if you did – be sure to change that up as well.

Almost done, need to do 1 more thing:

Activate (this also saves it)



(that's actually 2 things)

Now, you can test it and make sure you're good to go –

Open up Service Portal
Fill out Apply for Telework
If you change the 'Requested For' field does the manager change?

- If the manager is blank, it means the person doesn't have a manager listed
- Does the rest of the form work right?
- Submit Go to the Telework workspace
  Check the new telework form
- Is the manager field correct?
- Is the requested for field correct?
- After the approval, does the email go to the right person?

# That's it!!

If you need to change how variables respond, use the dev studio and there's always a place for scripting – whether it's just one line of code or something more complex that uses a script include, it's all there.

# Portal: Overview

## Purpose

Requestors have commented that they would like to see the SHIELD Agency's branding in the Service Portal and would like a way to track their Telework Case requests and edit them after submission.

To add branding to the Service Portal, we will be using the Branding Editor to make some simple changes to the look and feel of the application.

We will also modify the "Requests" page to show users all the Telework Case requests that they've submitted or have been assigned to them and add a button that will allow them to navigate to an editable view of the requests in their list.

# Notes

-

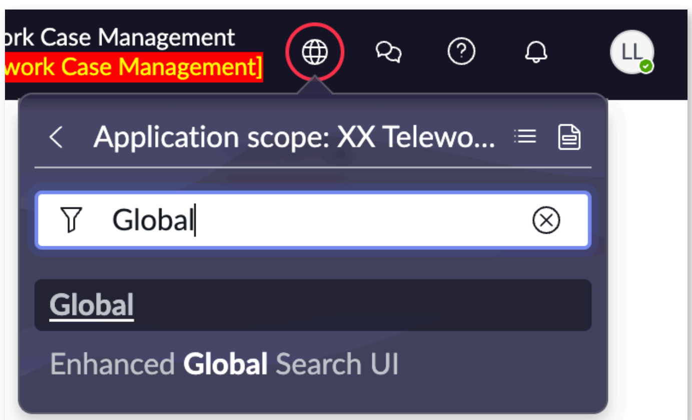
# Portal: Branding

After all the work we've done to enhance the functionality of the Telework Case Management application, it's finally time to make it pretty!

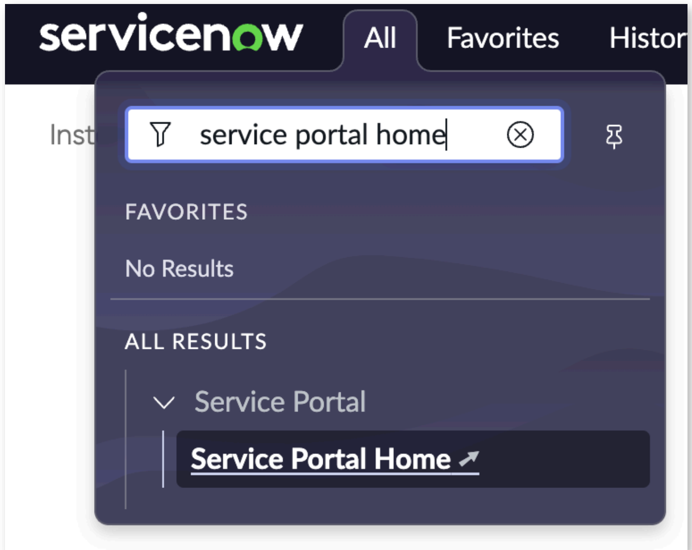
In this exercise, we will learn how to add the SHIELD Agency's logo and branding to the Service Portal using the Branding Editor.

## Steps

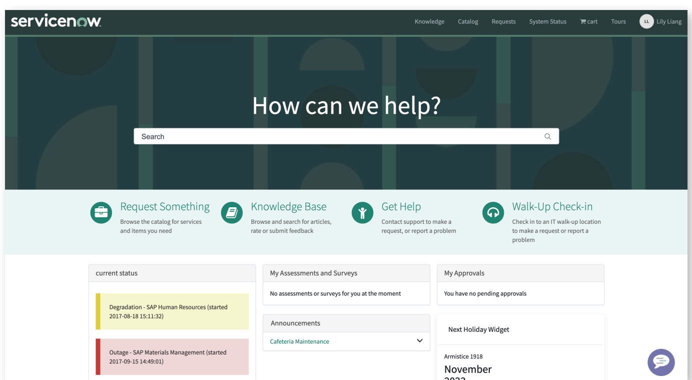
1. Change your scope to Global by choosing it from the Application picker in the top right of your screen in the platform view



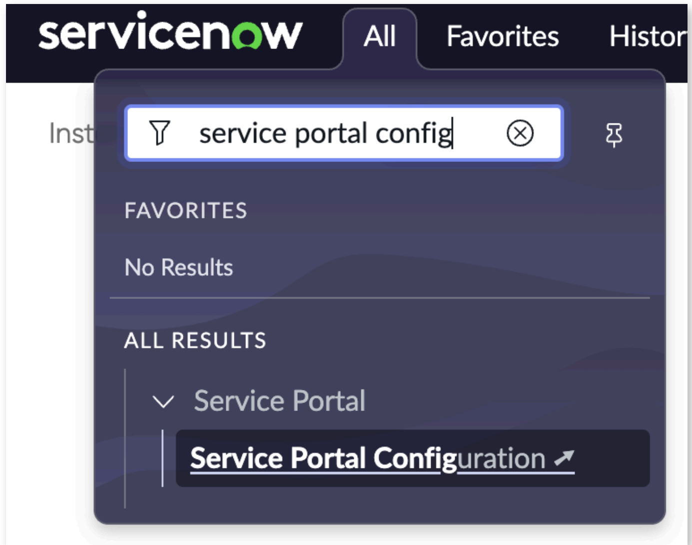

2. Take a look at the OOB portal by navigating to Service Portal > Service Portal Home



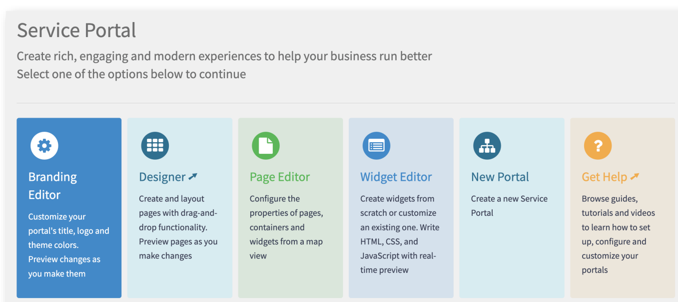

Your portal will look like this:



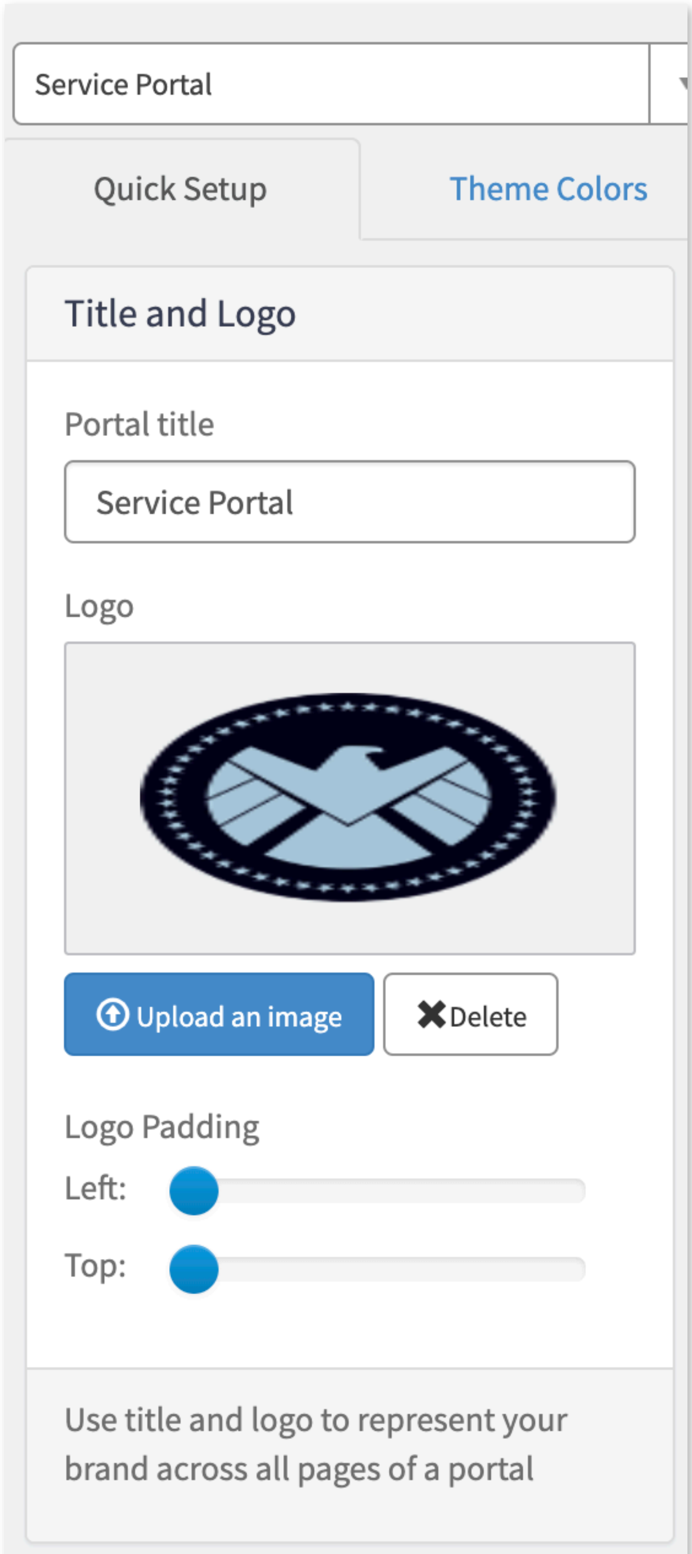

3. To access the Branding Editor, go back to the platform tab and navigate to Service Portal > Service Portal Configuration




4. Select Branding Editor




5. Change the logo by uploading an image and select SHIELD_Logo.png


Service Portal
Quick Setup
Theme Colors
Title and Logo
Portal title
Service Portal
Logo
Upload an image
Delete
Logo Padding
Left:
Top:
Use title and logo to represent your brand across all pages of a portal

6. Scroll down in the quick setup panel to change the **tag line color** and **background image.**
   Use Portal_Background.png as your background image.

   ```
   Tag line color: #62D84E
   ```

## Tag Line and Background

Tag Line

How can we help?

Tag Line color

#62D84E

Homepage background color

#fff

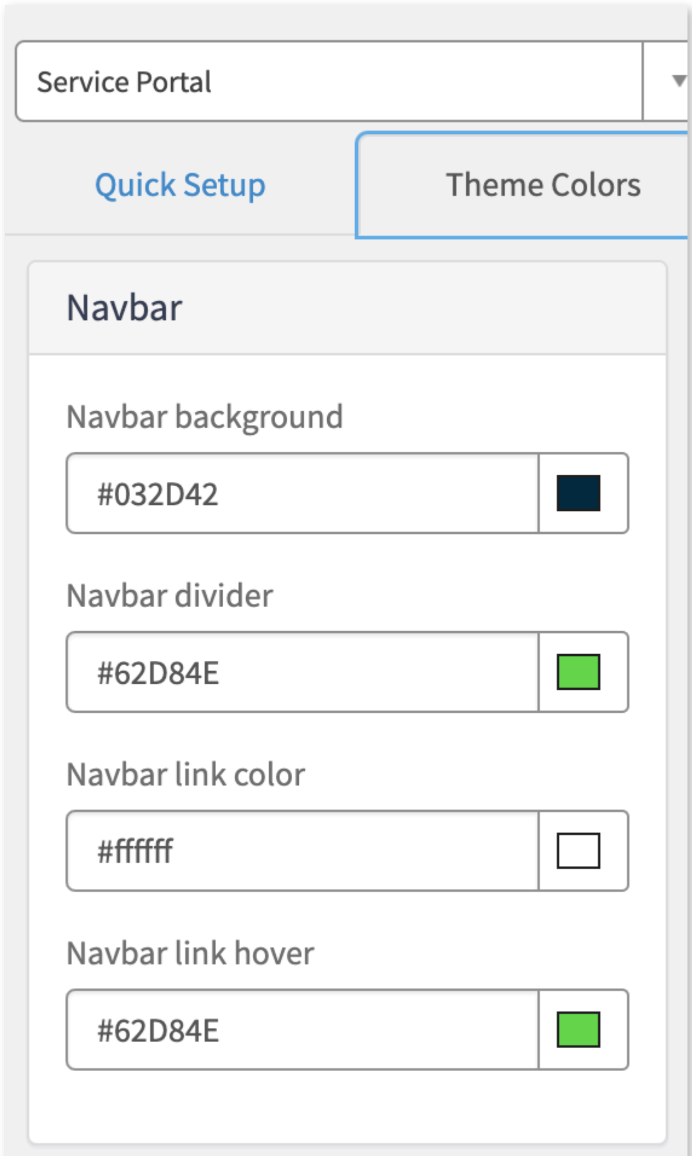
Background Image

Upload an image

Delete

Reinforce your company theme with company graphics and colors

7. Switch to the Theme Colors tab




8. Update the following values:
Navbar:

```
Navbar background: #032D42
Navbar divider: #62D84E
Navbar link color: #ffffff
Navbar link hover: #62D84E
```

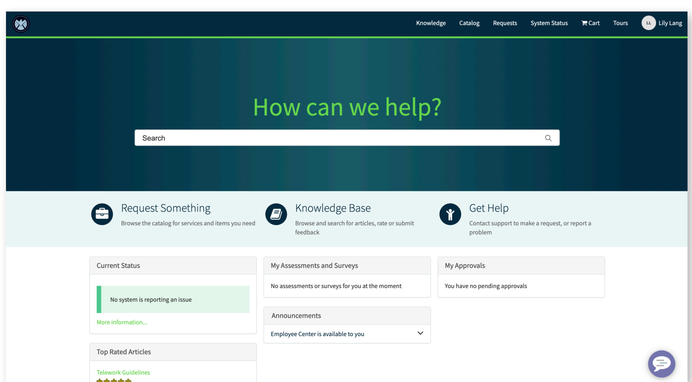
Brand:

```
Button default background: #fff
Primary: #032D42
```

Text:

```
Link color: #62D84E
```

9. Return to the platform view and navigate to **Service Portal** > **Service Portal Home** to check out your new portal!




# Notes

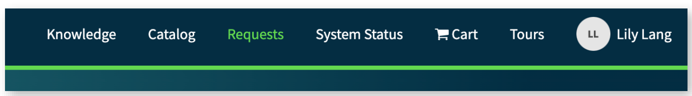
-

# Portal: Add a Widget

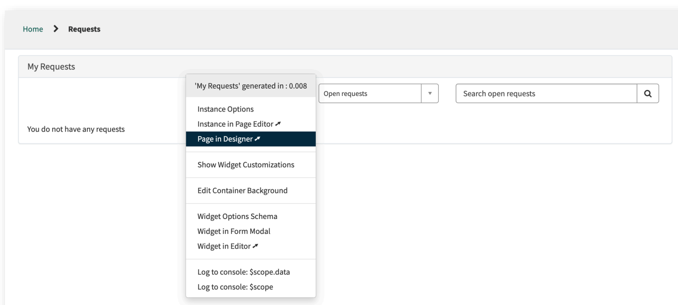
In this exercise, we will learn how to configure an existing page and widget to give users access to the list of Telework Case requests that they've submitted or are assigned to them.

## Steps

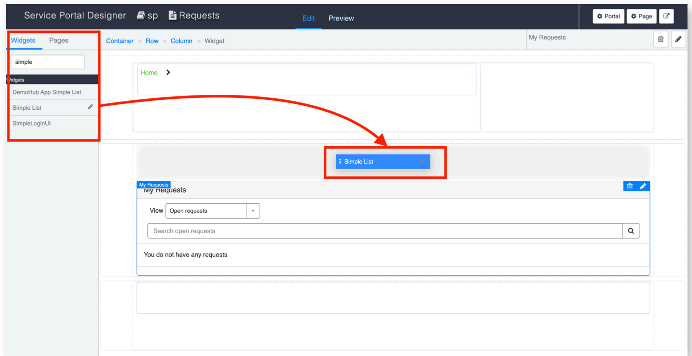
1. From the service portal home, select **Requests** in the navbar at the top right of the screen



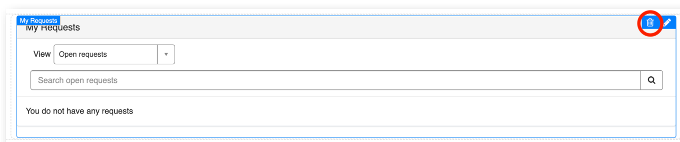

2. **Ctrl + Right Click** on the “My Requests” widget and select **Page in Designer**



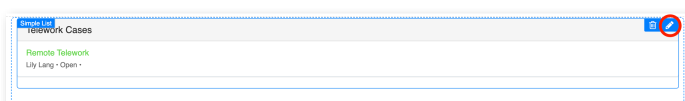

3. Find the **Simple List** widget in the list on the left of the page and **drag/drop** it above the “My Requests” widget




4. Remove the My Requests widget from the page by clicking the **trash can icon** on the top right of the widget



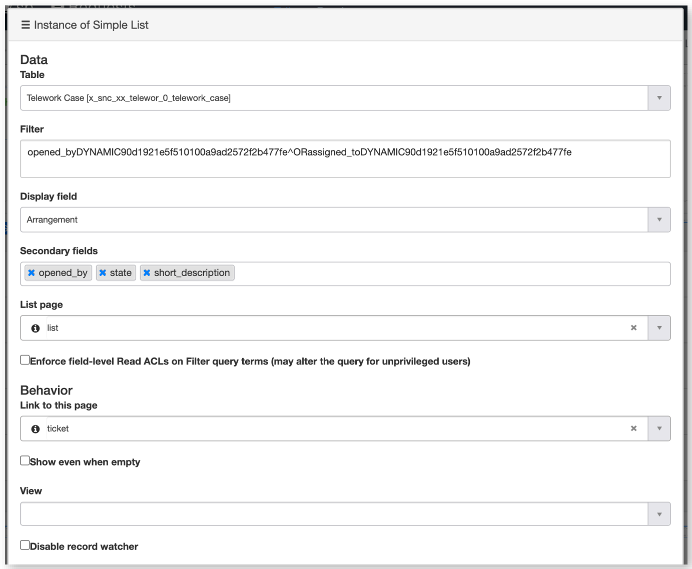

5. Configure the **widget options** for the Simple List by selecting the pencil icon on the top right of the widget



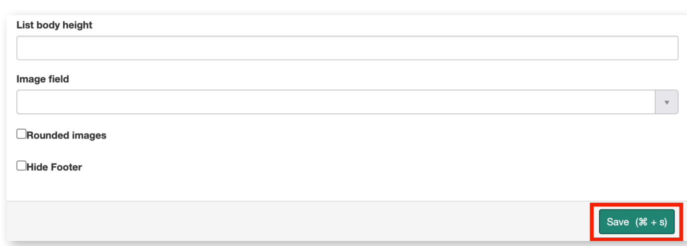

6. Set the following values:
**Table:** Telework Case
**Display Field:** Arrangement

**Secondary fields:** Opened by, State, Short Description
**List page:** list
**Link to this page:** ticket
**Filter:** *This will show all Telework Cases opened by or assigned to the logged in user*

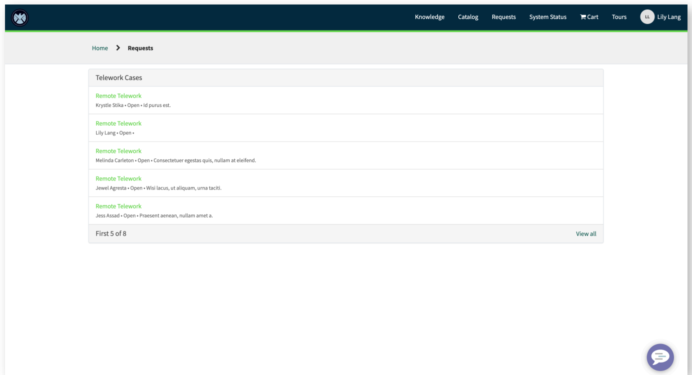
```
opened_byDYNAMIC90d1921e5f510100a9ad2572f2b477fe^ORassigned_toDYN
```



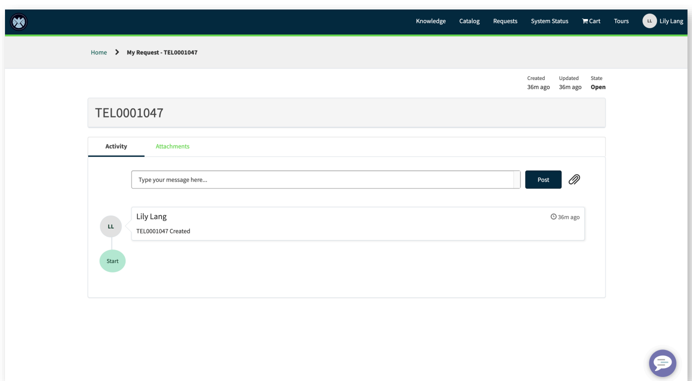

7. **Save** your widget options (you may need to scroll down)



8. Refresh the tab where your portal **request page** is open to see your new widget




9. Click on any record to navigate to the ticket page




# Notes

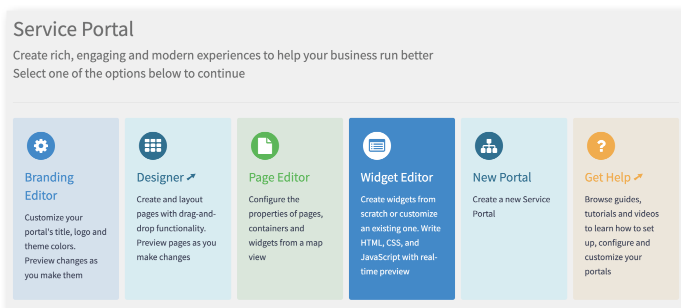
-

# Portal: Create a Widget

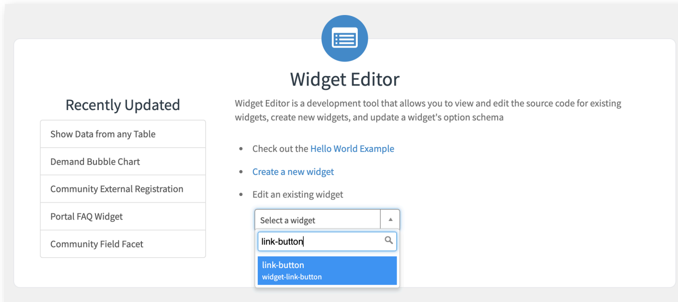
In this exercise, we will learn how to clone an existing widget so we can modify it. We will use the button to allow users to navigate to an editable view of the Telework Cases that they've submitted or are assigned to them.

## Steps

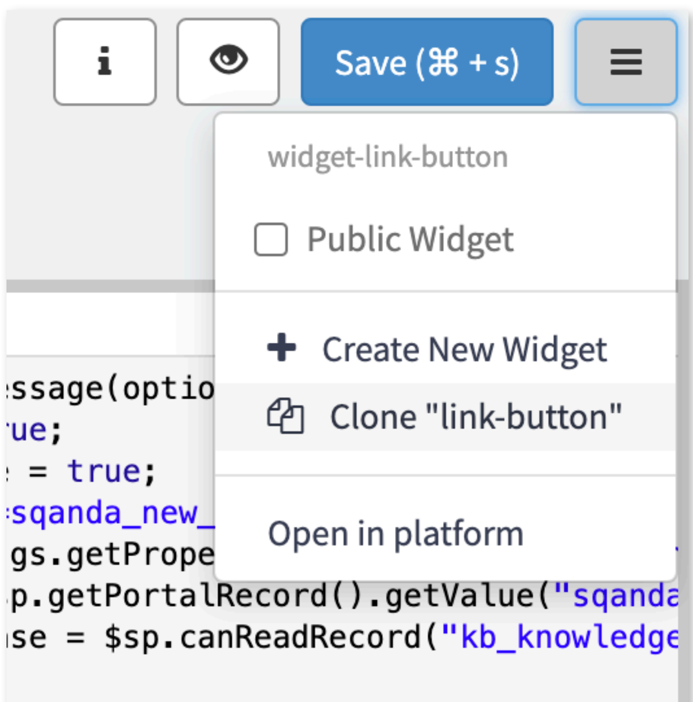
1. To access the Widget Editor, navigate to Service Portal > Service Portal Configuration, then click Widget Editor



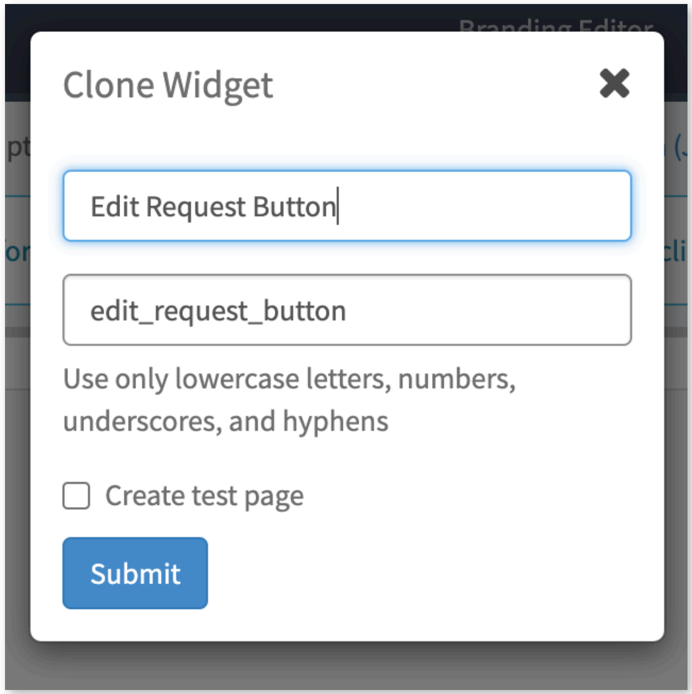

2. Select link-button under Edit an existing widget




3. Select **Clone "link-button"** from the additional options from the menu in the top right

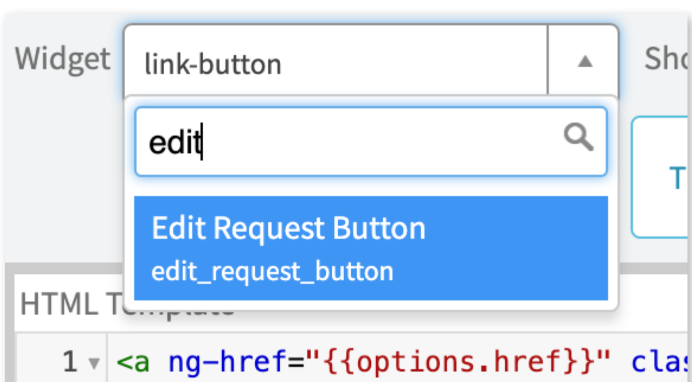



4. Name your widget **Edit Request Button**. The Widget ID will auto-populate




5. Submit

6. Select your new widget in the top left of the widget editor

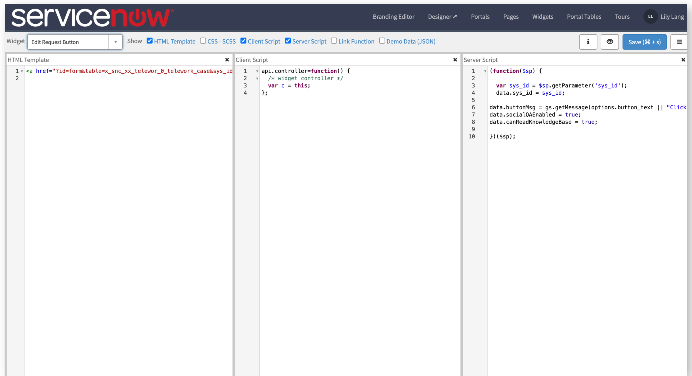


7. Replace the HTML Template and Server Script with the following code:
HTML: *button will link to the "form" page and passes the table name and sys_id of the current record as parameters in the URL*

```
<a
href="?id=form&table=YOUR_TABLE&sys_id={{data.sys_id}}&view=sp"
class="btn btn-{{options.color}} m-b">{{data.buttonMsg}}</a>
```

Server Script:

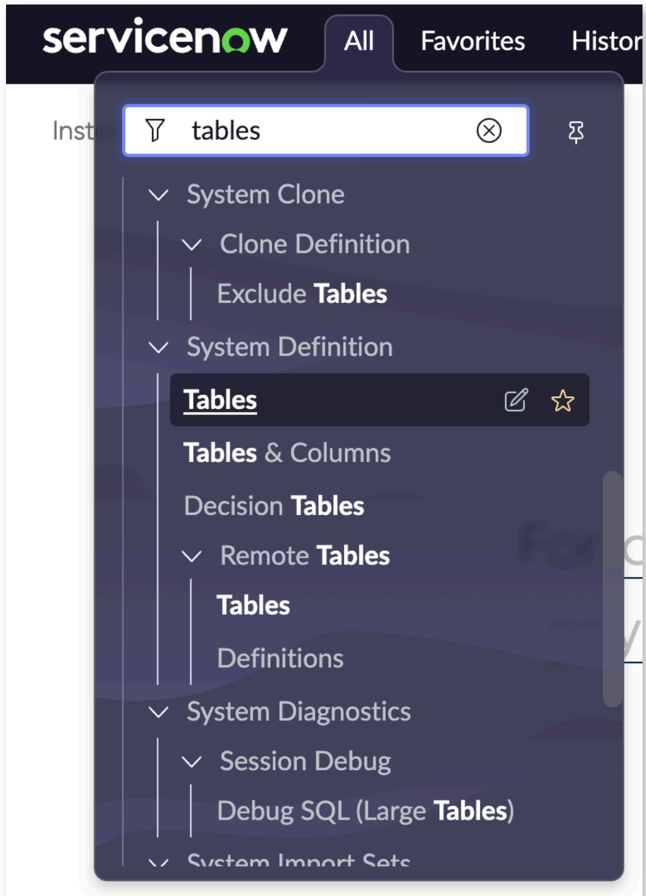
```
(function($sp) {

//get the sys_id from the current record page
url and adds it to the widget's data object so
it can be accessed from HTML
var sys_id = $sp.getParameter('sys_id');
data.sys_id = sys_id;

//set the text that shows up in the button to
either the message configured through the
widget options or defaults to "Click Here" if
nothing is configured
data.buttonMsg =
gs.getMessage(options.button_text || "Click
Here");

})($sp);
```



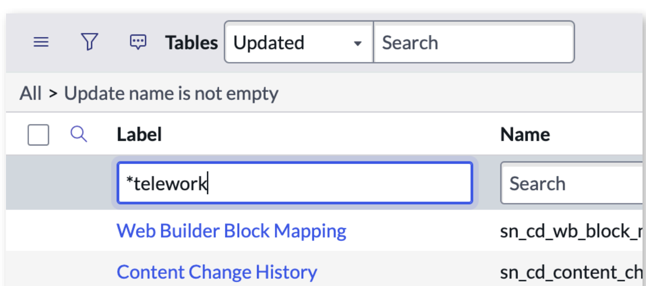

8. **Save** using the blue button at the top right of the widget editor

9. Replace **YOUR_TABLE** in the HTML template with the name of your table

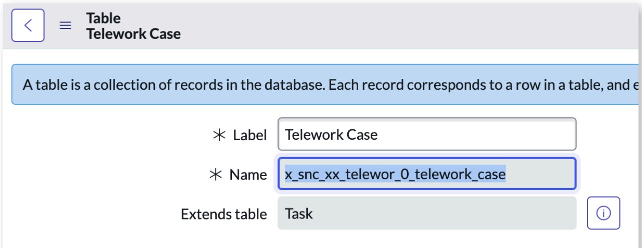
    - In the platform view, navigate to **System Definition** > **Tables**



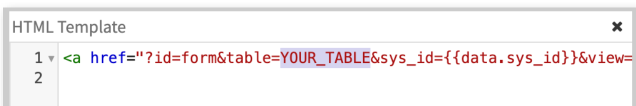

- In the search bar below "Label" search for: *telework



- Hit **Enter/Return** on your keyboard
- Open your **Telework Case** table by clicking on the label
- Copy the value in the **Name** field

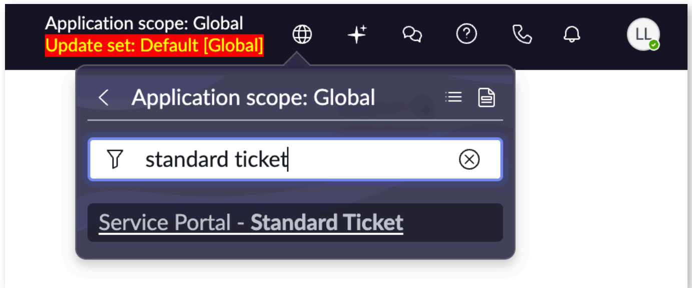



- Go back to the browser tab where the widget editor is open and replace **YOUR_TABLE** with the copied value

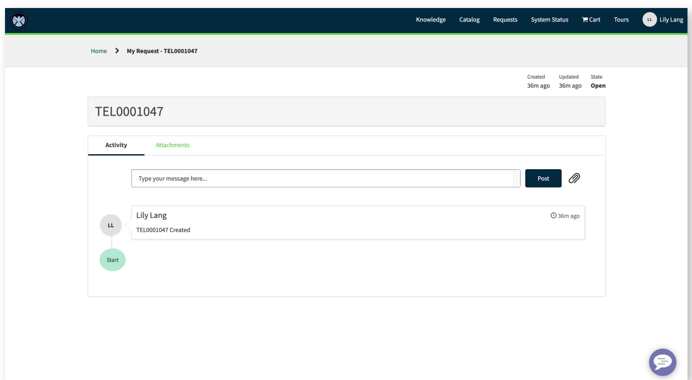



- Save

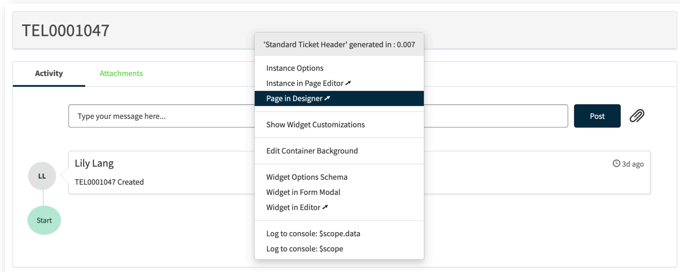
10. Go back to the platform view and change your scope to **Service Portal – Standard Ticket** using the scope picker on the navbar



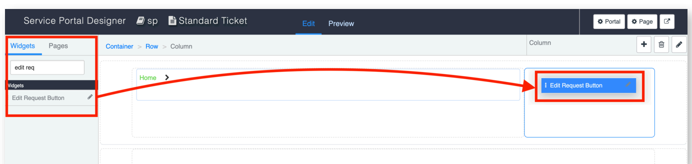

11. Navigate back to the Service Portal request page and click on any telework case to take you to the **ticket page**



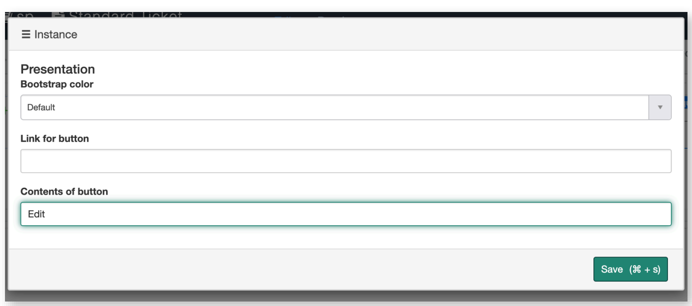

12. **Ctrl + Right click** on the widget and select **Page in Designer**




13. Drag and drop your Edit Request Button widget to the top right section of the page

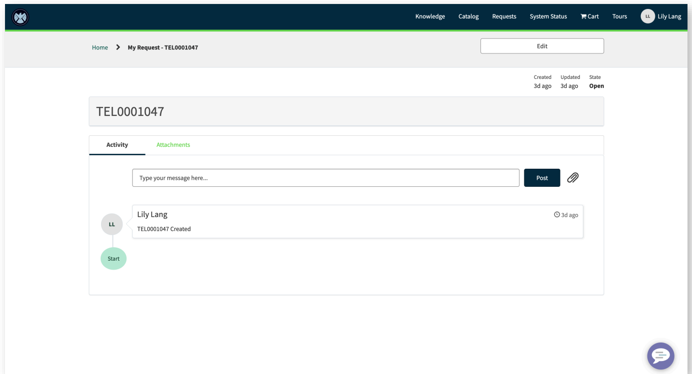



14. Configure the options for your widget and set the Contents of button to **Edit**

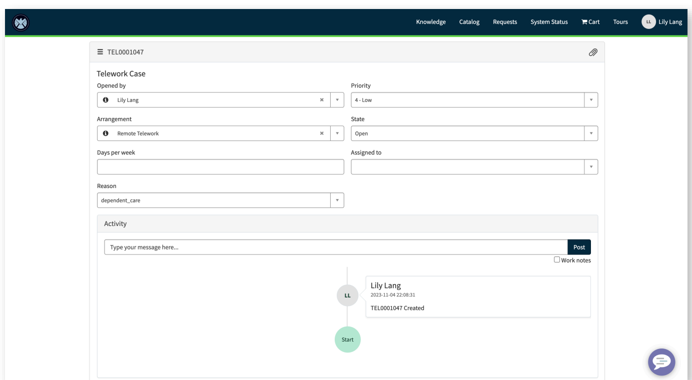



15. Save

16. Navigate back to the tab where your ticket page is open and **reload** the page to see your new button




17. Use the Edit button to see your ticket in an editable form page




# Notes

- 

# Conclusion

In this lab, we built, implemented, and tested an application using the fundamental part of an application: Data, User Experience, and Automation.

The team managing the work will be more efficient and the users will get what they need faster.

Of course, the examples in these exercises are just the beginning when it comes to the power of leveraging Automation Engine.

The possibilities are limited only by your imagination as a Creator. If you can dream it, you can workflow it on the Now Platform.

Here are some resources to help you continue your journey on the ServiceNow platform

# We Value Your Feedback!

## Help Us Improve Your Learning Experience

As we wrap up our Telework 201 workshop, we would greatly appreciate your input. Your feedback is vital in shaping future workshops to better meet your needs and expectations. We invite you to take a few moments to complete our post-workshop survey.

### Why Your Feedback Matters:

- Enhance Workshop Quality: Your insights help us understand what works well and what can be improved, ensuring that future workshops are even more engaging and effective.
- Tailor Content to Your Needs: Your suggestions play a crucial role in helping us tailor our content to better suit your learning preferences and professional requirements.
- Foster a Community of Learning: By sharing your thoughts, you contribute to a community that values growth, innovation, and collaborative learning.

### Confidential and Anonymous:

Your responses will be kept confidential and anonymous. We are interested in your honest opinions, whether they are positive or negative.

Click on the link below to access the survey. It should take no more than **5 minutes** to complete.

Start the Survey

Thank you for taking the time to help us enhance the Telework 201 workshop experience.

# CHANGELOG

## History

| Version | Author | Changes |
|---|---|---|
| 2023-November (Vancouver) | Marc.Mouries(ServiceNow) | Initial release |

# Download Lab Guide PDF

Click here to download Lab Guide PDF