Release Process Management

ServiceNow - App Engine Workshop

Allianz Technology - München

02.08.2022 10:30 – 15:30

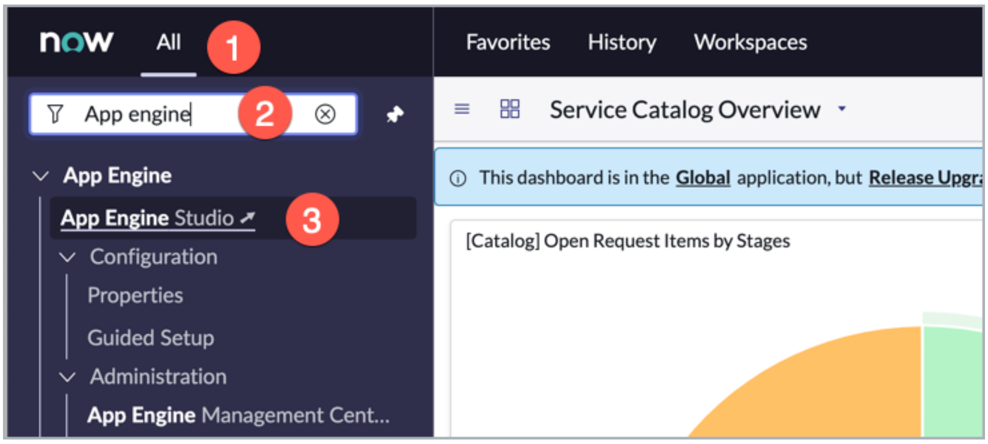
PLEASE SEND YOUR FEEDBACK TO THIS LAB GUIDE TO:
Dirk.Redeker@ServiceNow.com

Table of Contents

1 Prepare your Dev environment

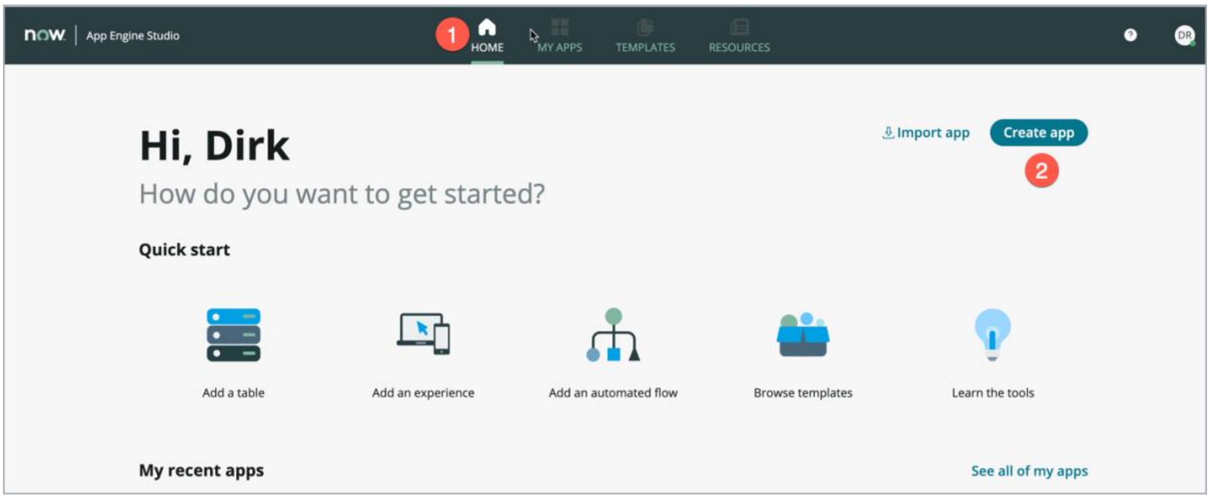
1.1 Exercise #1 – Create the Application

In the top menu bar, click "**All**" **(1)** and enter "**App engine**" **(2)** to the filter navigator. From the search results in the navigator, select "**App Engine Studio**" **(3)**.



App Engine Studio opens in a new Browser tab.

Make sure, the "**Home**" Tab **(1)** is selected and then click "**Create app**" **(2)**



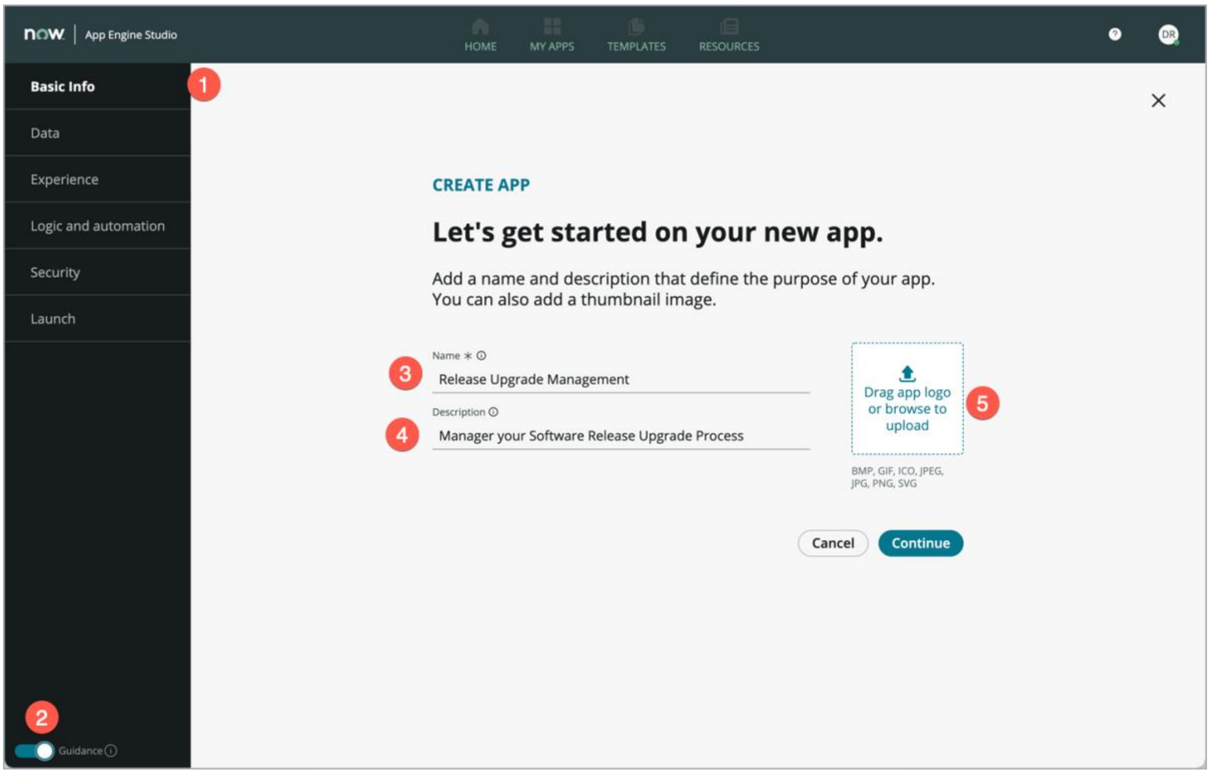
On the first screen that opens (see below), review the sidebar (1) showing your current location. You can hide the side bar by clicking the toggle at the left bottom of the screen (2).

Enter a meaningful "**Name**" (3) for your App. In this case name it "**Release Upgrade Management**" and optionally enter a "**Description**" (4).
You can also upload a picture as the logo for your App (5), which is used in the overview of Apps on the App Engine Studio Home page. Uploading a Logo is an optional step, but it really looks nice in the App Overview.

In the next screenshot, you can see that I have clicked the toggle in the lower-left edge to hide the side panel (2) – giving me some more space on the screen.

You can also add a logo for your App (5), by uploading an image file.

Click "**Continue**" to move on.



On the next screen, you can create and manage the Roles to be used in your App.
App Engine Studio already prepared two Roles for you.

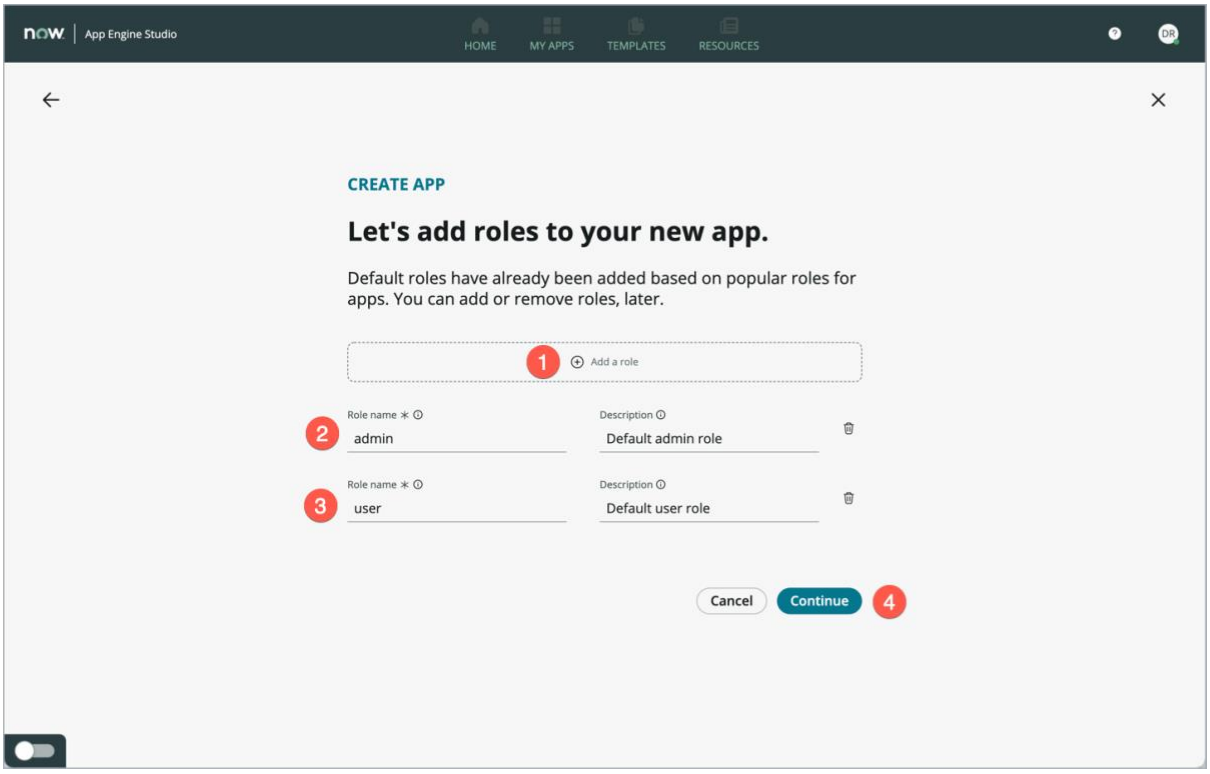
The "admin" Role (2) is intended to manage the settings and base data of the App, later in the production environment. Mostly, users with the "admin" Role also have access to all data.

Note:
This is an App-specific "admin" Role and is NOT the same as the "System Administrator" (admin) Role!

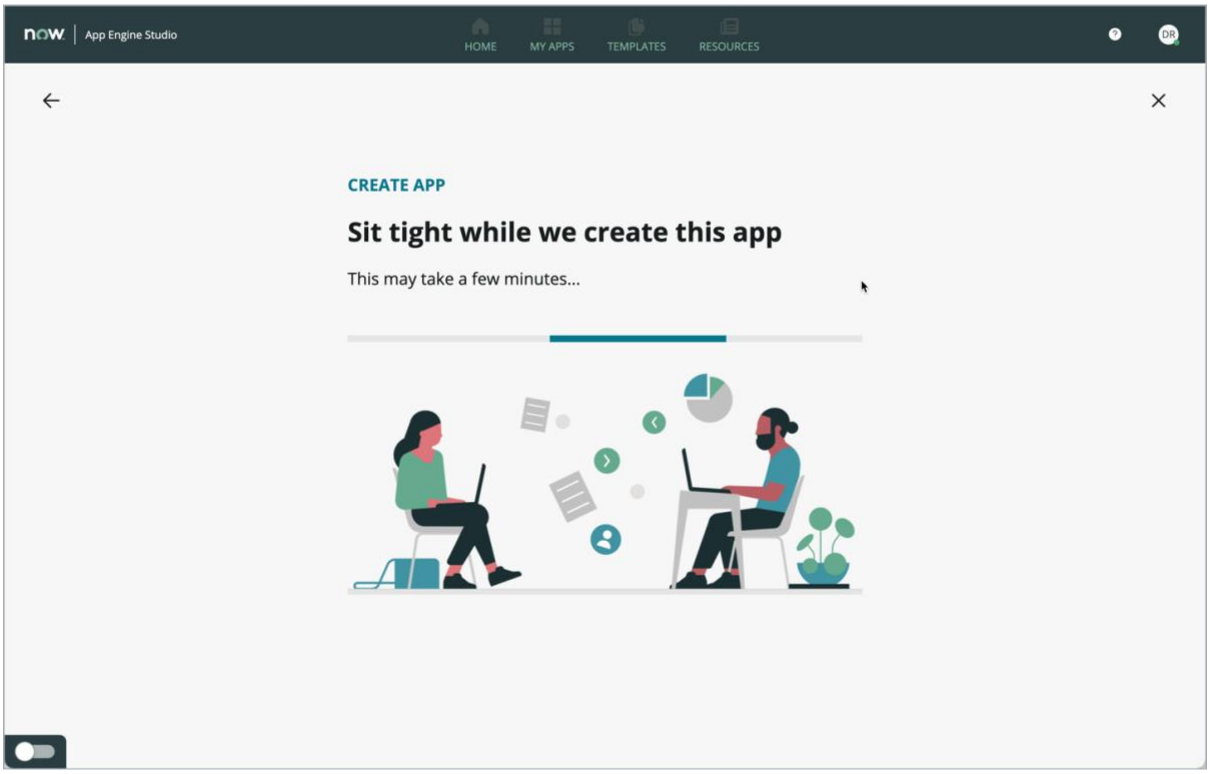
The “user” Role (3) is intended to use all or parts of the applications functions. Think about the Persona that your App is addressed to. It may make sense to create (at least) one Persona specific Role for your App.

You can add more new custom Roles to your App (which makes sense in the context of your App) by clicking “Add a role” (1)

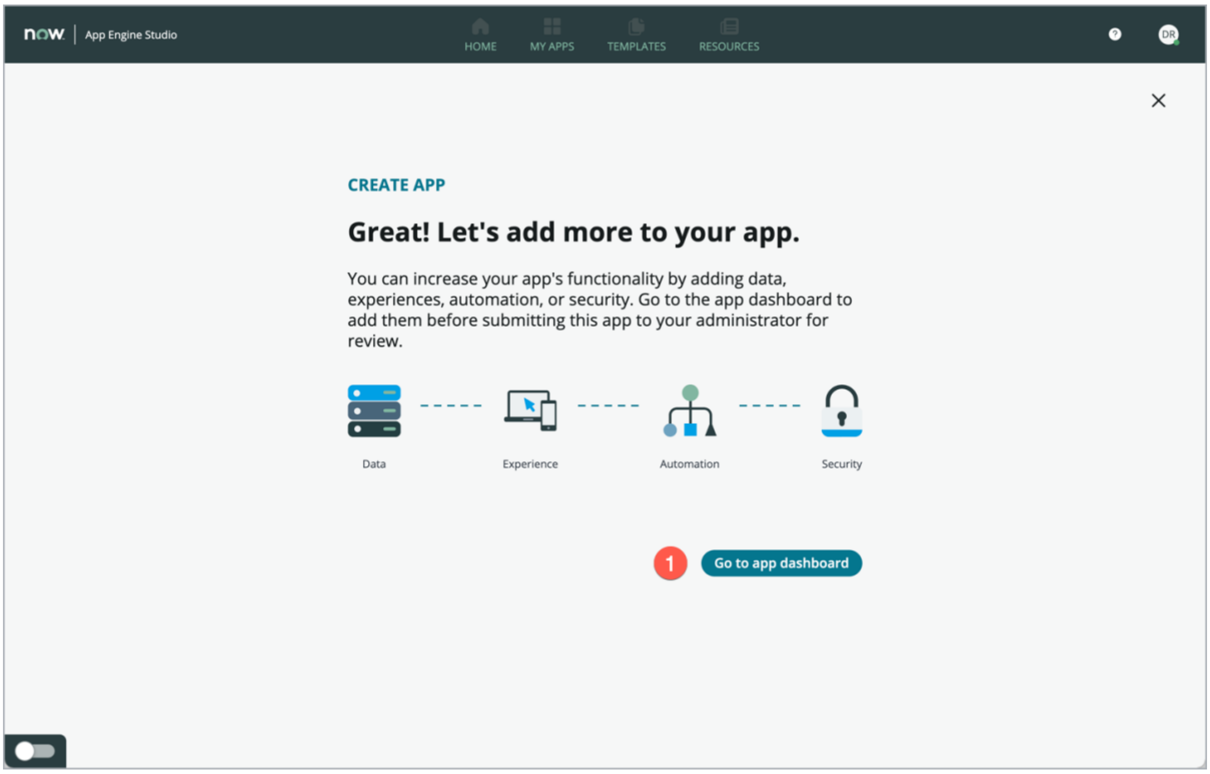
For our Lab, we do not focus on this topic, so we will keep the predefined Roles.
Click “Continue” (4) to move on.



App Engine Studio now creates a new App, which also gets its own (so called) Scope in ServiceNow. Scopes isolate Apps from each other, and Apps also allow to be deployed in separate packages during the Release cycle of your Application.



After a few moments, the new App is created in your ServiceNow instance and you can now navigate to the App’s Dashboard by clicking “Go to app dashboard” (1).

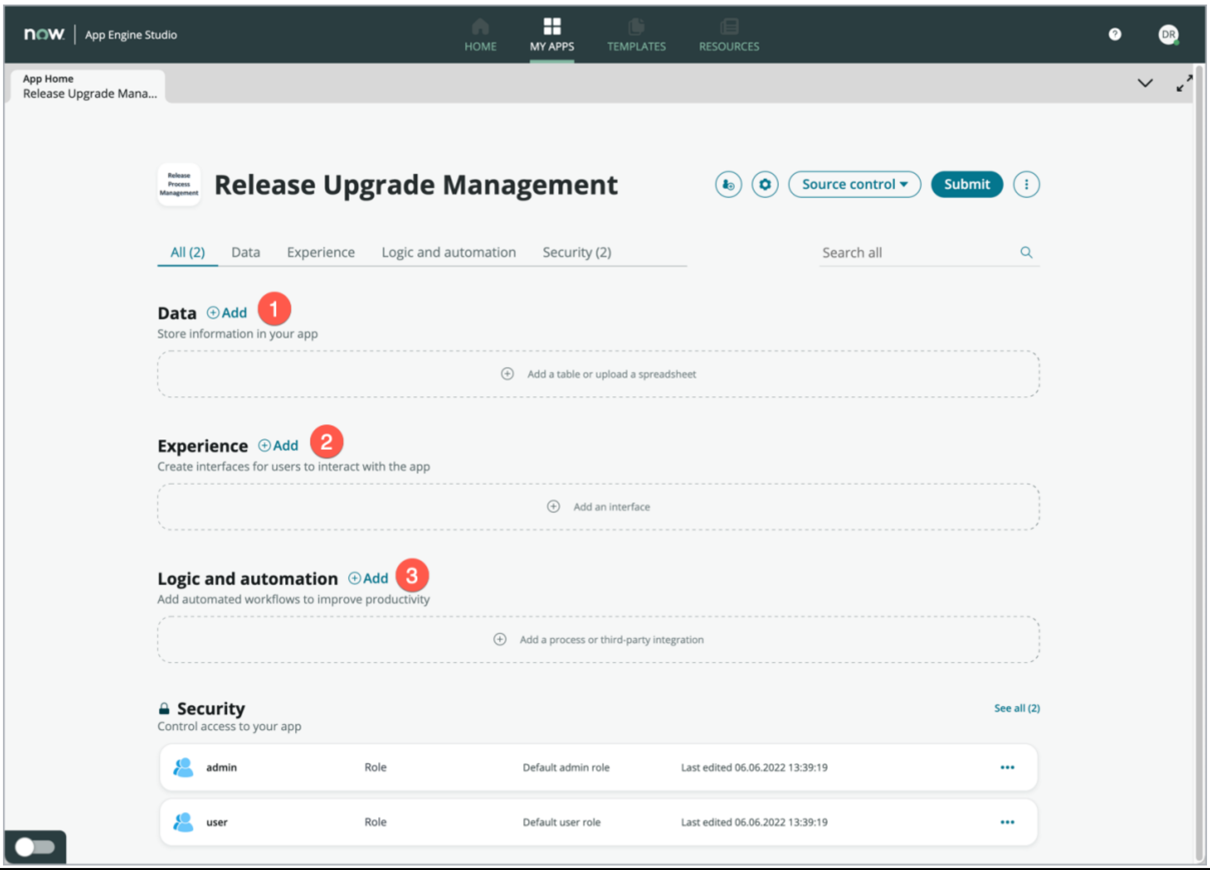


Welcome to the dashboard of your Application in App Engine Studio.
For this Lab, we will mainly focus on the three areas indicated below.

In the “**Data**” section, you can find “add” (1) and manage the database tables that are created as part of this Application.

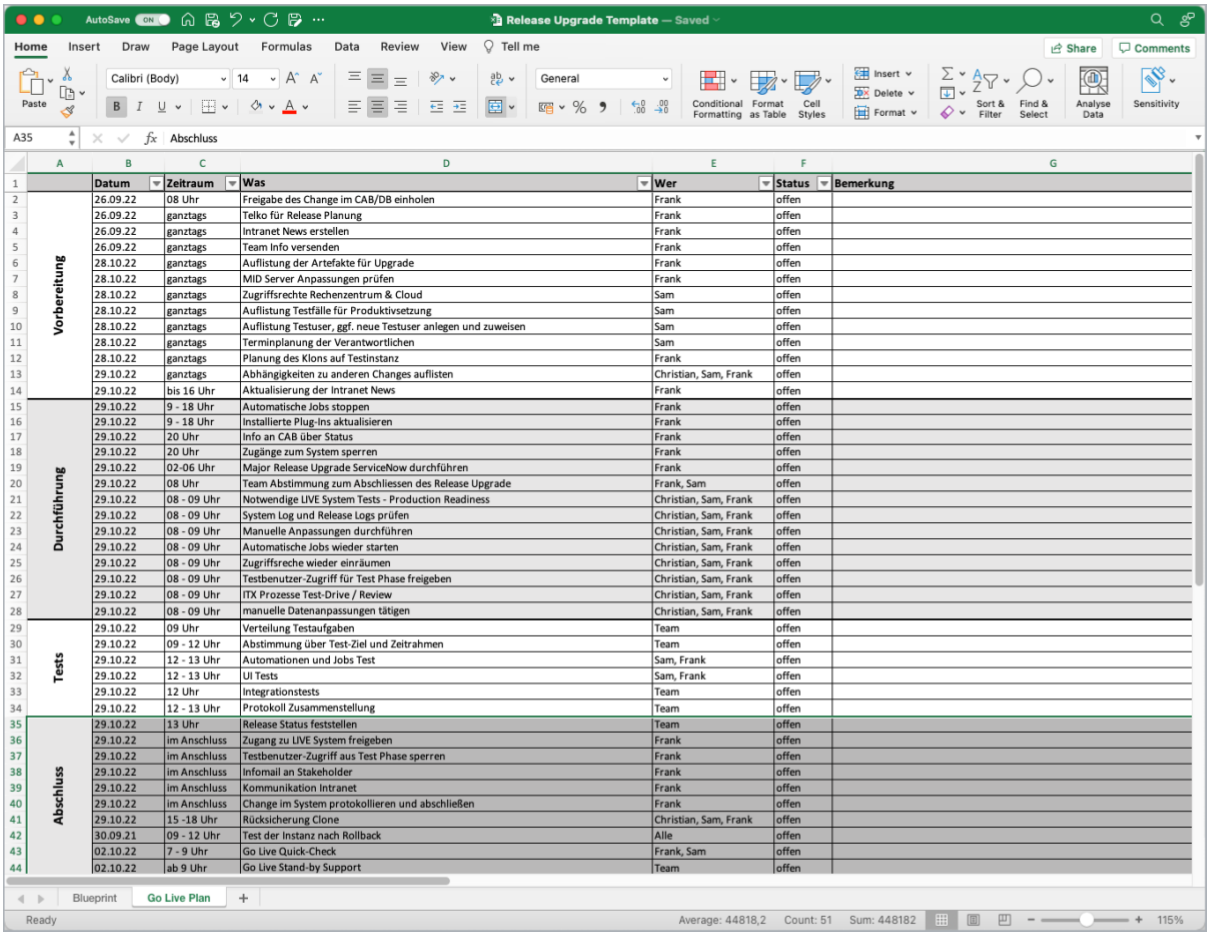
In the “**Experience**” section, you can find “add” (2) and manage all User interfaces that are created as part of this Application.

In the “**Logic and automation**” section, you can find “add” (3) and manage all Computer interfaces. Automation and integration which is part of this Application.



2 Data and tables

Today, the process is completely run using an Excel file, which you can see in the screenshot below:



	Datum	Zeitraum	Was	Wer	Status	Bemerkung
Vorbereitung	26.09.22	08 Uhr	Freigabe des Change im CAB/DB einholen	Frank	offen	
	26.09.22	ganztags	Telko für Release Planung	Frank	offen	
	26.09.22	ganztags	Intranet News erstellen	Frank	offen	
	26.09.22	ganztags	Team Info versenden	Frank	offen	
	28.10.22	ganztags	Auflistung der Artefakte für Upgrade	Frank	offen	
	28.10.22	ganztags	MID Server Anpassungen prüfen	Frank	offen	
	28.10.22	ganztags	Zugriffsrechte Rechenzentrum & Cloud	Sam	offen	
	28.10.22	ganztags	Auflistung Testfälle für Produktivsetzung	Sam	offen	
	28.10.22	ganztags	Auflistung Testuser, ggf. neue Testuser anlegen und zuweisen	Sam	offen	
	28.10.22	ganztags	Terminplanung der Verantwortlichen	Sam	offen	
	28.10.22	ganztags	Planung des Klons auf Testinstanz	Frank	offen	
	29.10.22	ganztags	Abhängigkeiten zu anderen Changes auflisten	Christian, Sam, Frank	offen	
	29.10.22	bis 16 Uhr	Aktualisierung der Intranet News	Frank	offen	
Durchführung	29.10.22	9 - 18 Uhr	Automatische Jobs stoppen	Frank	offen	
	29.10.22	9 - 18 Uhr	Installierte Plug-Ins aktualisieren	Frank	offen	
	29.10.22	20 Uhr	Info an CAB über Status	Frank	offen	
	29.10.22	20 Uhr	Zugänge zum System sperren	Frank	offen	
	29.10.22	02-06 Uhr	Major Release Upgrade ServiceNow durchführen	Frank	offen	
	29.10.22	08 Uhr	Team Abstimmung zum Abschliessen des Release Upgrade	Frank, Sam	offen	
	29.10.22	08 - 09 Uhr	Notwendige LIVE System Tests - Production Readiness	Christian, Sam, Frank	offen	
	29.10.22	08 - 09 Uhr	System Log und Release Logs prüfen	Christian, Sam, Frank	offen	
	29.10.22	08 - 09 Uhr	Manuelle Anpassungen durchführen	Christian, Sam, Frank	offen	
	29.10.22	08 - 09 Uhr	Automatische Jobs wieder starten	Christian, Sam, Frank	offen	
	29.10.22	08 - 09 Uhr	Zugriffsreche wieder einräumen	Christian, Sam, Frank	offen	
	29.10.22	08 - 09 Uhr	Testbenutzer-Zugriff für Test Phase freigeben	Christian, Sam, Frank	offen	
	29.10.22	08 - 09 Uhr	ITX Prozesse Test-Drive / Review	Christian, Sam, Frank	offen	
	29.10.22	08 - 09 Uhr	manuelle Datenanpassungen tätigen	Christian, Sam, Frank	offen	
Tests	29.10.22	09 Uhr	Verteilung Testaufgaben	Team	offen	
	29.10.22	09 - 12 Uhr	Abstimmung über Test-Ziel und Zeitrahmen	Team	offen	
	29.10.22	12 - 13 Uhr	Automationen und Jobs Test	Sam, Frank	offen	
	29.10.22	12 - 13 Uhr	UI Tests	Sam, Frank	offen	
	29.10.22	12 Uhr	Integrationstests	Team	offen	
	29.10.22	12 - 13 Uhr	Protokoll Zusammenstellung	Team	offen	
Abschluss	29.10.22	13 Uhr	Release Status feststellen	Team	offen	
	29.10.22	im Anschluss	Zugang zu LIVE System freigeben	Frank	offen	
	29.10.22	im Anschluss	Testbenutzer-Zugriff aus Test Phase sperren	Frank	offen	
	29.10.22	im Anschluss	Infomail an Stakeholder	Frank	offen	
	29.10.22	im Anschluss	Kommunikation Intranet	Frank	offen	
	29.10.22	im Anschluss	Change im System protokollieren und abschließen	Frank	offen	
	29.10.22	15 -18 Uhr	Rücksicherung Clone	Christian, Sam, Frank	offen	
	30.09.21	09 - 12 Uhr	Test der Instanz nach Rollback	Alle	offen	
	02.10.22	7 - 9 Uhr	Go Live Quick-Check	Frank, Sam	offen	
	02.10.22	ab 9 Uhr	Go Live Stand-by Support	Team	offen	

This Excel file is used for exactly ONE Release Upgrade process. That means, that we will need a new Excel file for each Release Upgrade Process. In fact, the Excel file is set up for the PROCESS of one Release and contains a number of Tasks that need to be worked on to reach the target (to Release the Upgrade successfully in time).

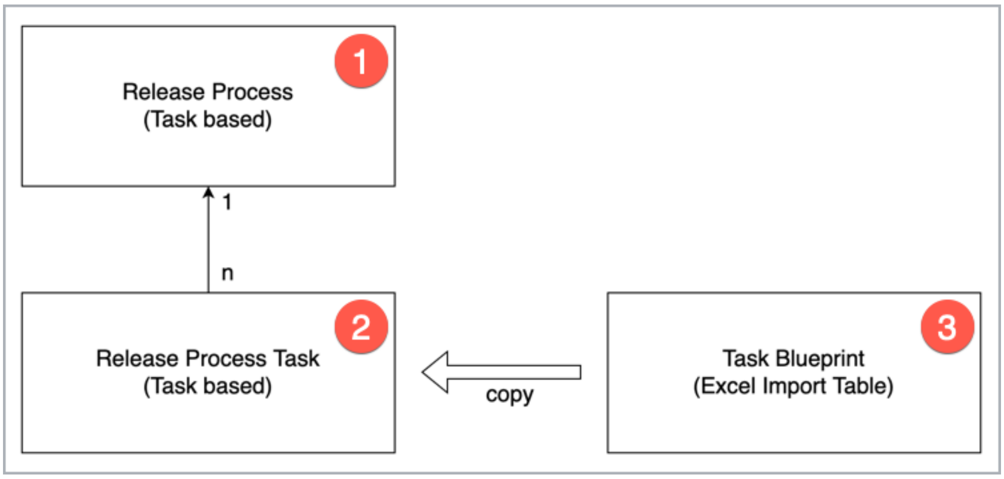
For the App, that we are going to build, we want to make it possible to drive any number of Upgrades, because Upgrades happen periodically (e.g. bi-annually like for ServiceNow).

Let's review the data structure, that we want to use for the App. We will work with three database tables.

The first table "**Release Process**" (**1**) is the level of the today's Excel file itself. There will be ONE data row for each Release Upgrade Process, which will be managed by the App.

The second table "**Release Process Task**" (**2**) is the level of the Tasks inside today's Excel file, where all separate Tasks are listed and assigned to individuals for fulfillment. The arrow from the "Release Process Task" up to the "Release Process" indicates, that each one "1" Release Process can contain any number of "n" Release Process Tasks.

Because each Release Process for a given App may follow the same (or at least a similar) schema, we do not want to capture all Tasks for each Release Process from scratch.
Therefore, we create a "**Task Blueprint**" (**3**) table, which holds the steps (tasks), which need to be used for each Release Process.
For the App, we want to allow to make a copy of the list of Blueprint Tasks to new "Release Processes" and use them as a template.



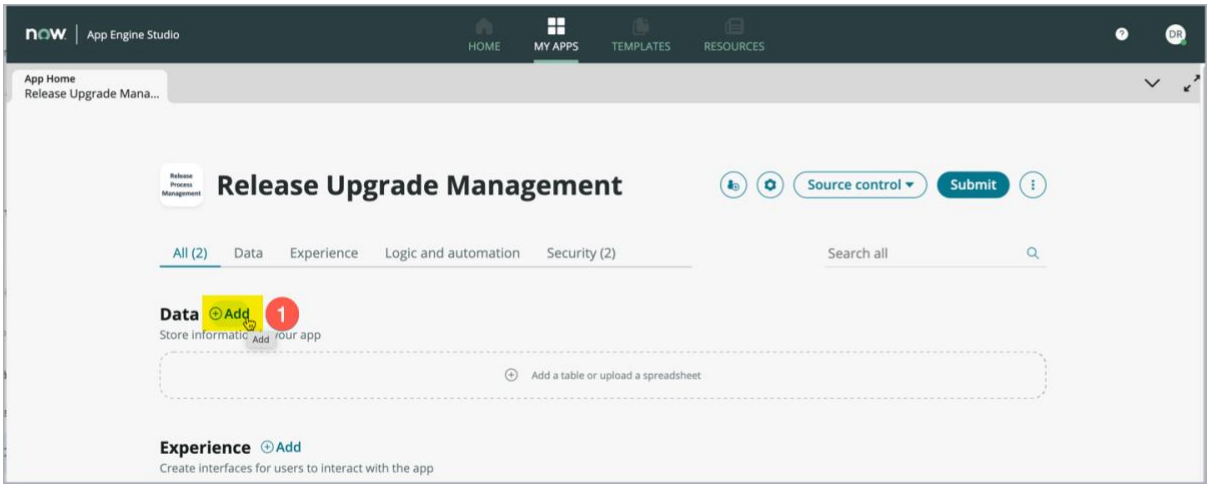
2.1 Exercise #2 – Create the database tables

In this exercise, we will create the table "**Release Process**" and "**Release Process Task**" shown in the diagram above.

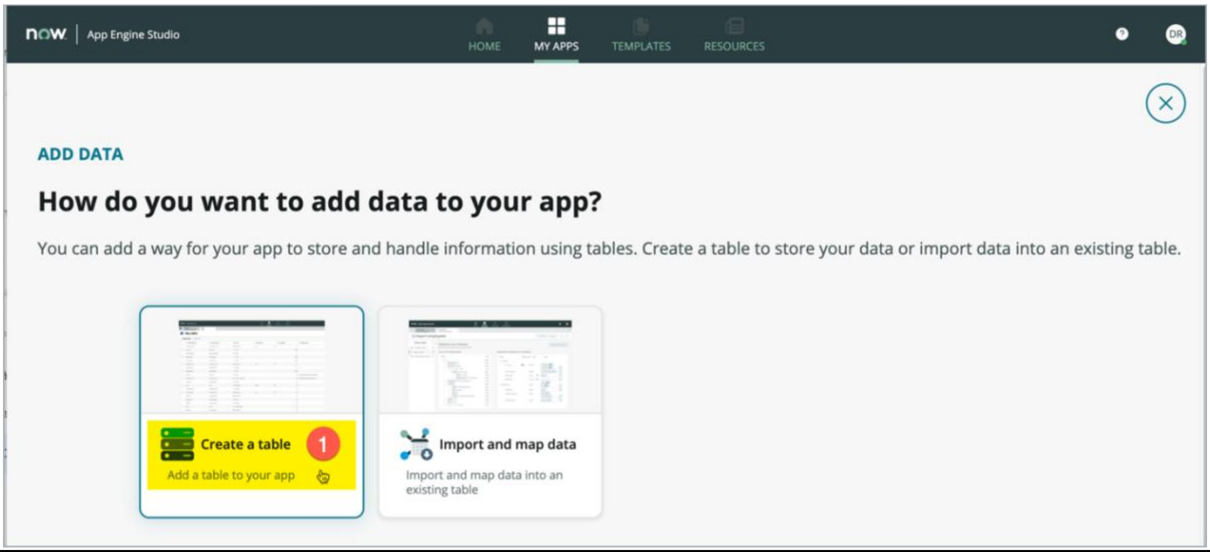
We will create both as tables **extended from the "task" table**. We will see later that there are great benefits by extending tables, and we will also review the benefits from using the main and very important "task" table in ServiceNow.

2.1.1 Create the "Release Process" table

On the Dashboard of your App, click "Add" (1) in the "Data" section.

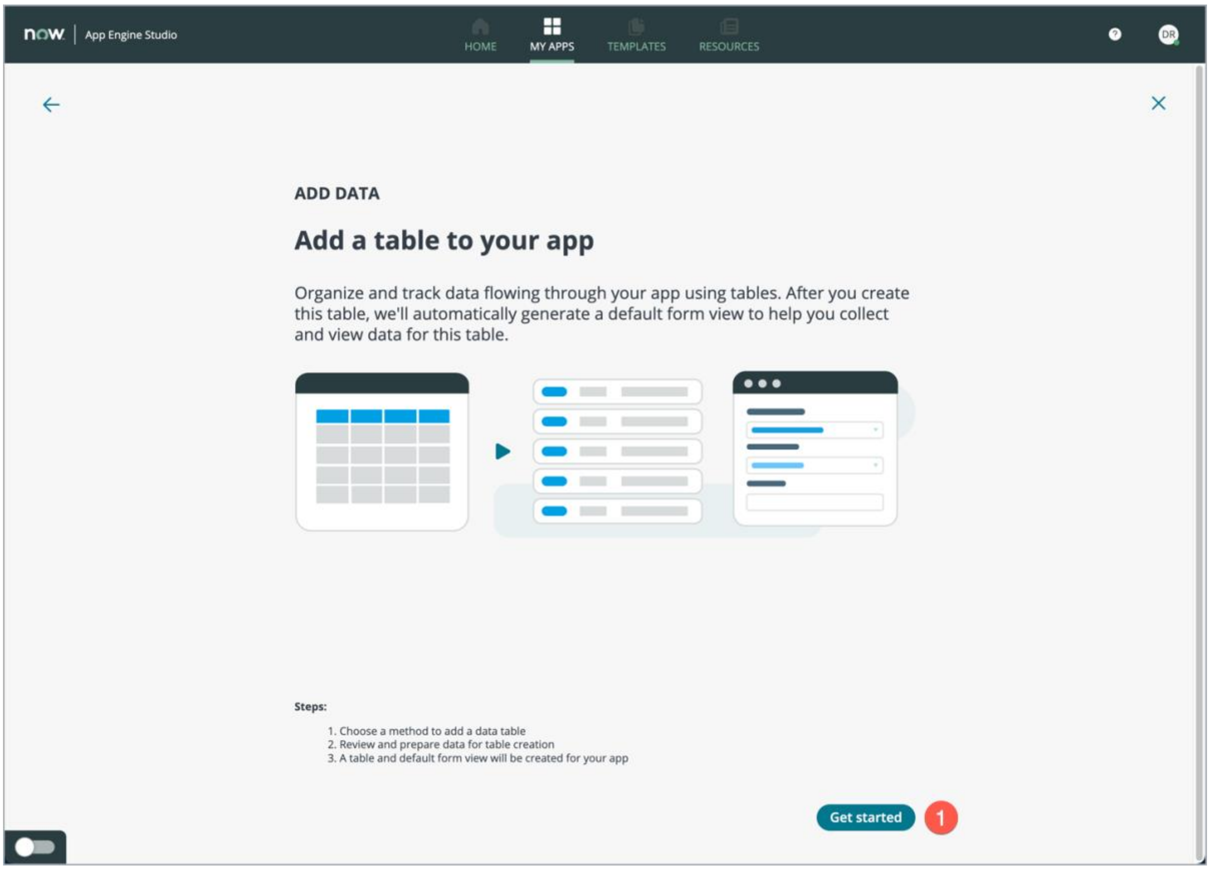


Select the left card "Create a table" (1).



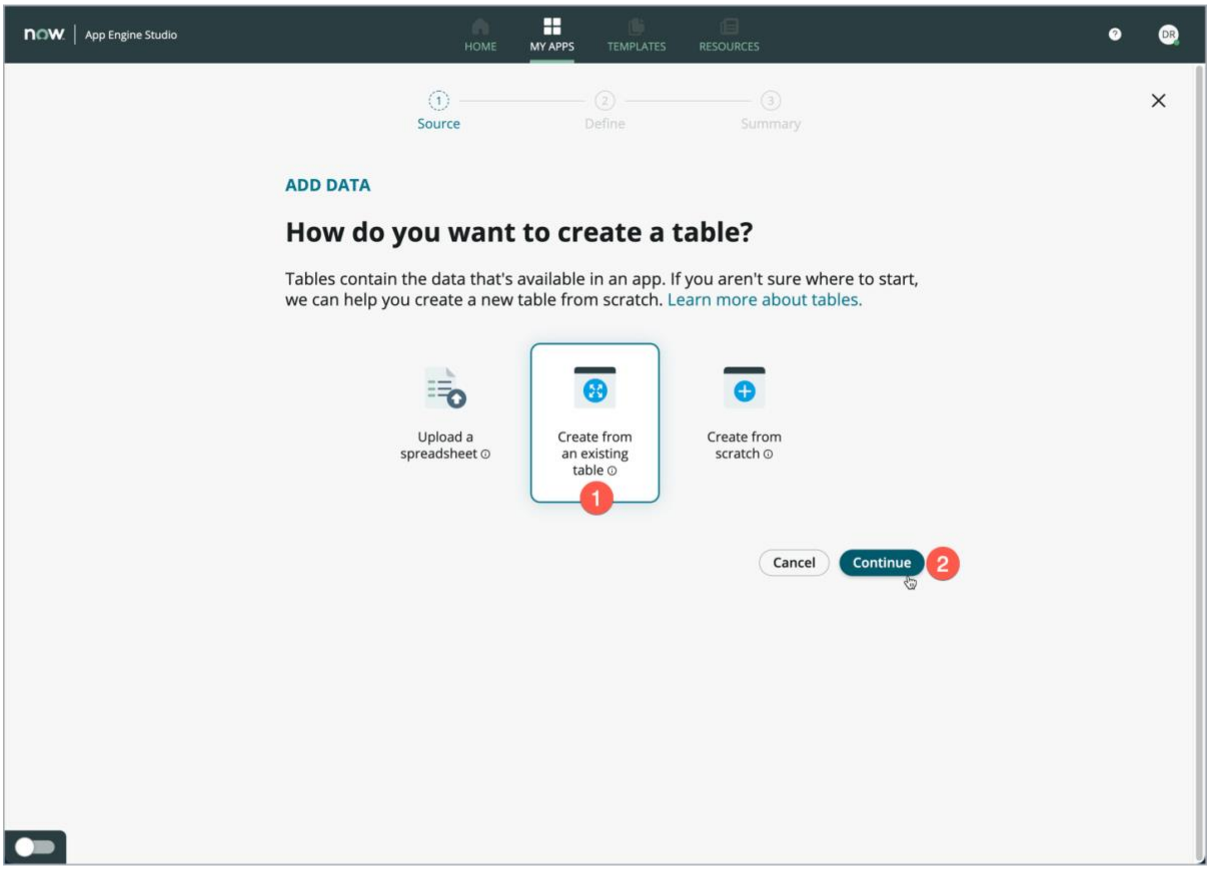
On the next screen, click “**Get started**” (1).

Note: If the “Get started” Button is not visible, scroll down the screen a bit.



Now, select the middle card “**Create from an existing table**” (1) and click “Continue” (2).

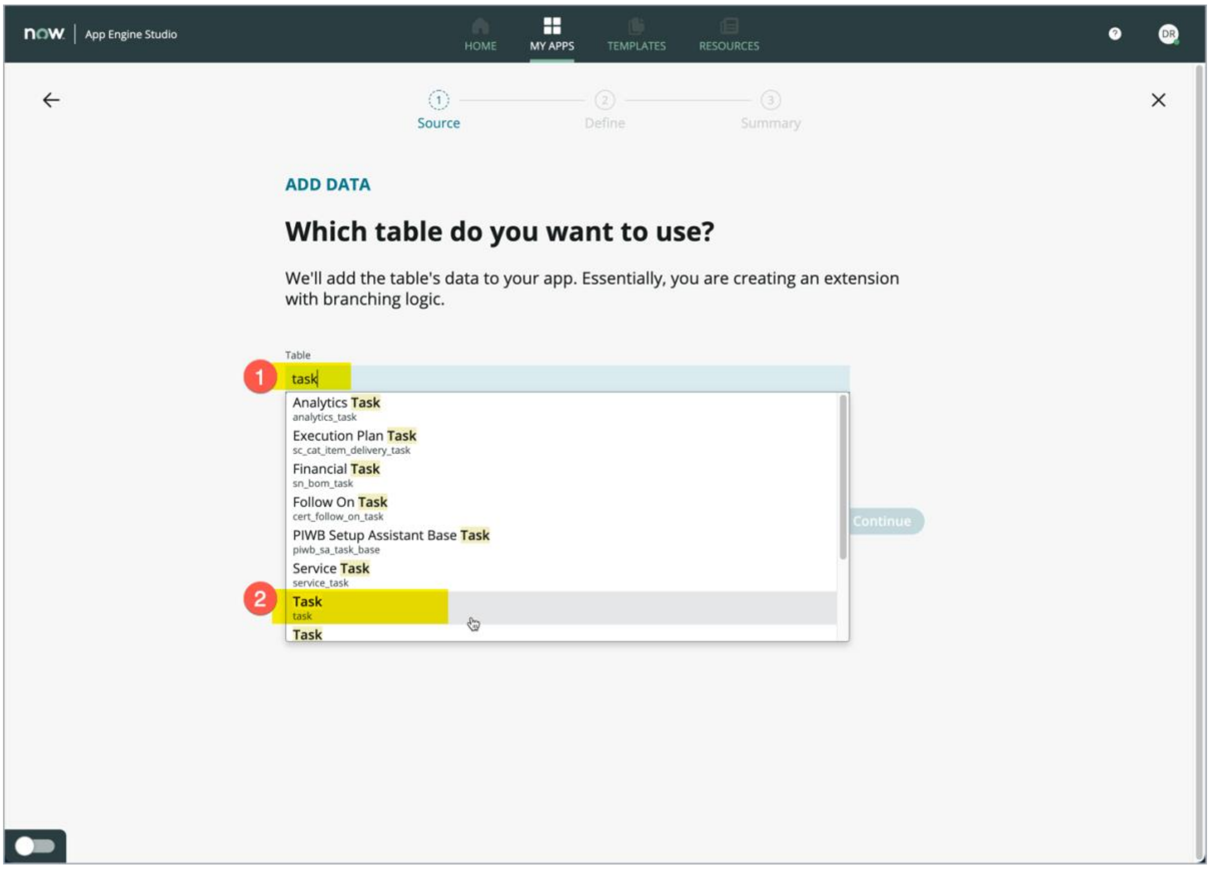
This option is used to build logic on top of existing database tables and its associated logic.



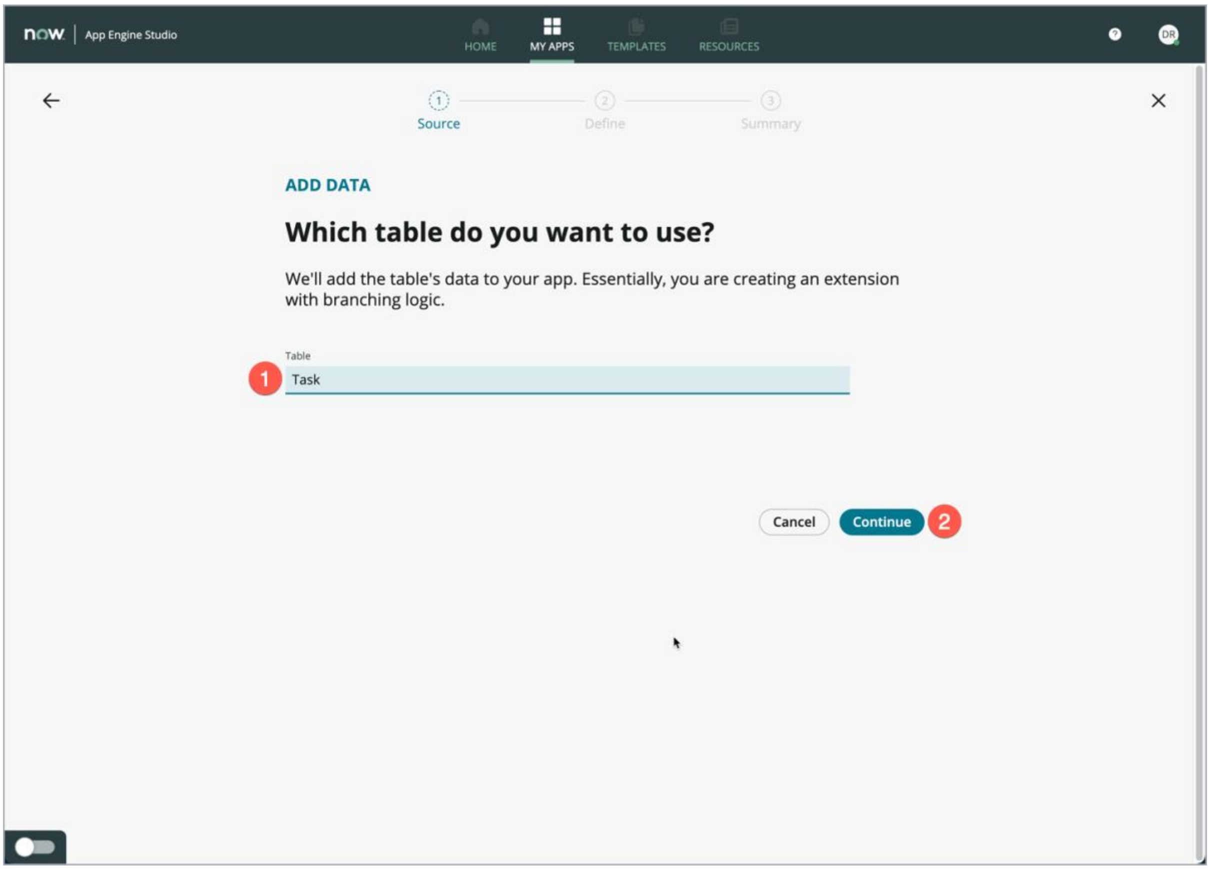
In the next step, you can select which table will be the base to be extended for your new table. Click the “**Table**” field (1) to open the dropdown and type “task” (1) in the blue field above the dropdown. This filters the list of shown tables to all tables containing “Task” in their name.

Note:
Scroll down if needed and select the “Task” table (**with the small “task”** mentioned below – see screenshot).

Note:
There are a lot of tables having “Task” in their name, but it is important for this lab to use the “Task” table as shown in the screenshot below.



After you selected the “Task” (1) table, click “Continue” (2) to move on.



Now, you need to name your table.

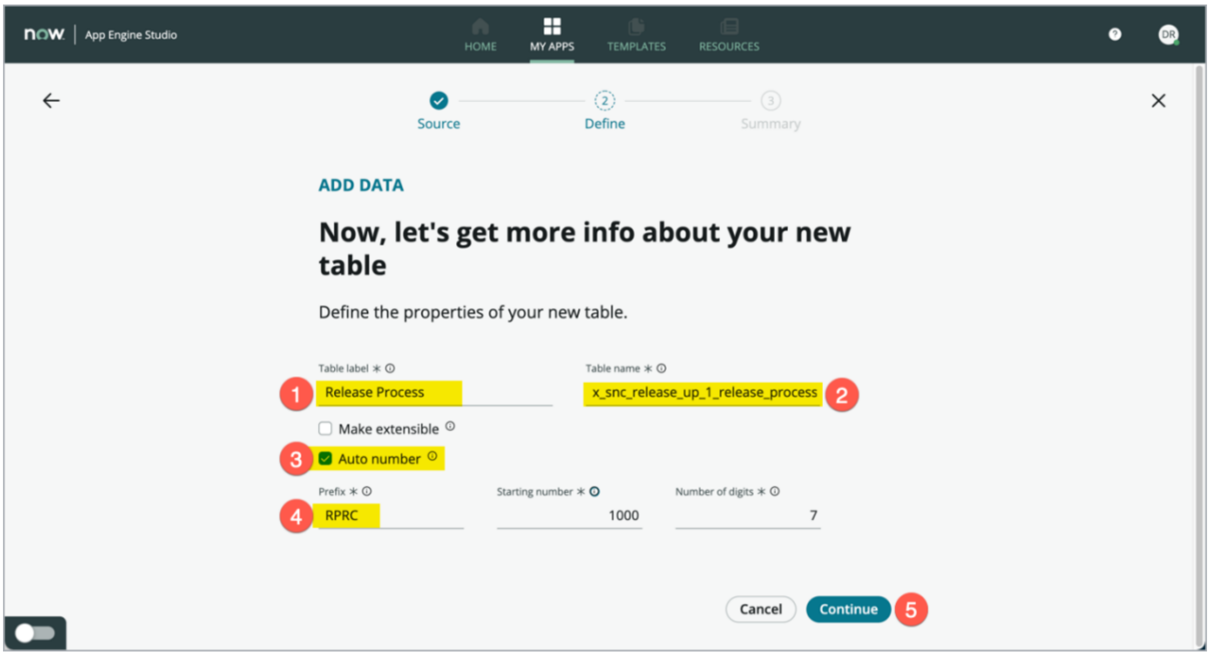
Enter the name "**Release Process**" in the "**Table label**" field (1), which is the name of the table shown to end-users. As soon as you leave this field, the "**Table name**" field (2) will be filled in automatically (that is the technical name of the table).

Note:
Do NOT change the "**Table name**" (2) field and just leave it as is!

Mark the "**Auto number**" Checkbox (3) and enter a valid "Prefix" (4) to be put in front of your Auto number. Use "**RPRC**" in this case.

Note:
Maybe a red annotation appears stating that the Prefix is invalid. In this case, change to focus of the field to (e.g.) "Table label" and review, if the red annotation disappears. If it does not disappear, change the value of the "Prefix" field and try again.

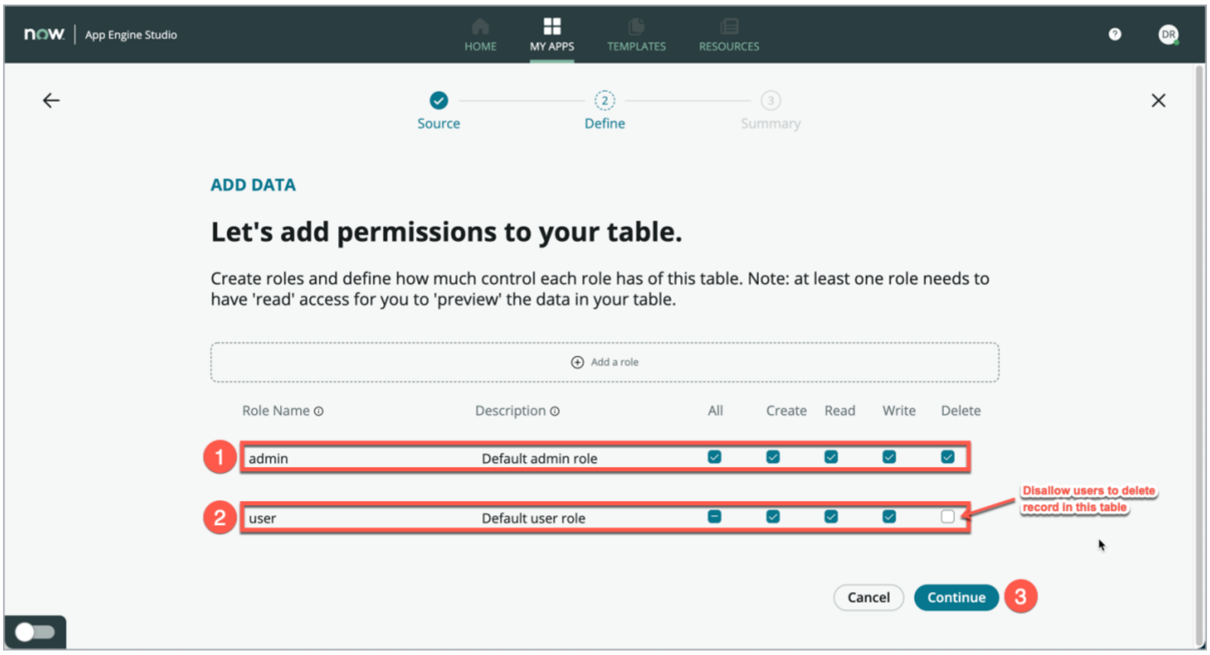
If all fields are filled in correctly, click "**Continue**" (5) to move on.



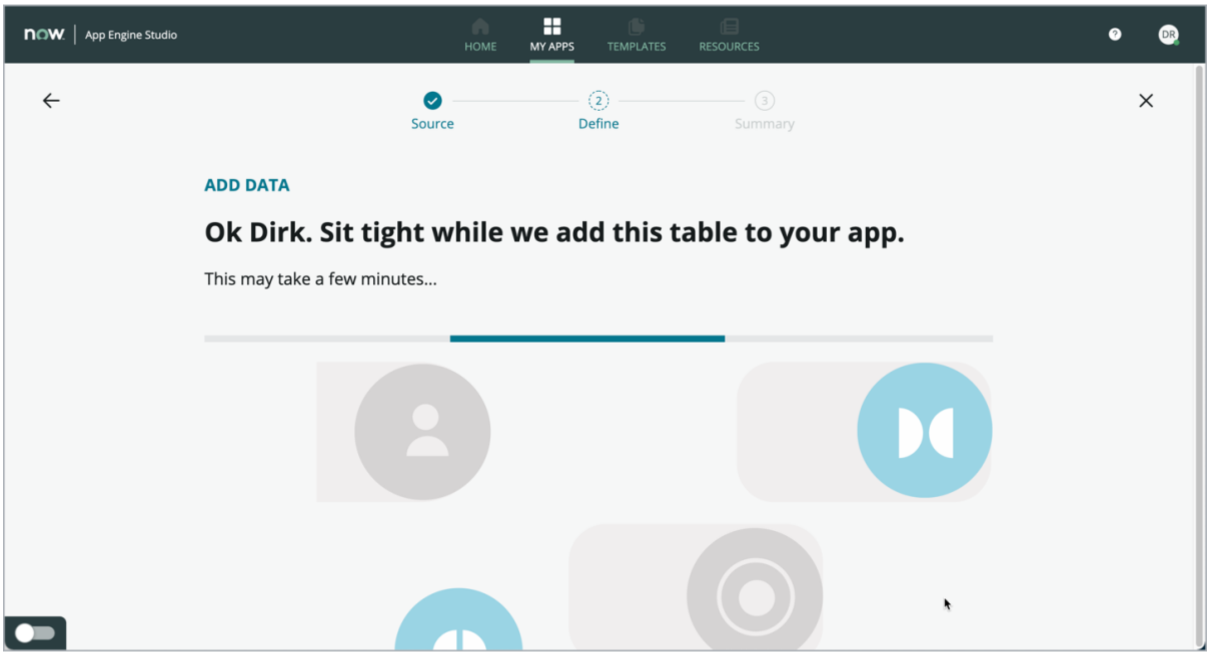
In the next step, you can configure which Roles of your Application will get which access to that new table.

For the “**admin**” Role, select **all checkboxes** (1) and for the “**user**” Role, select all checkboxes (2), **except the “Delete” checkbox**.

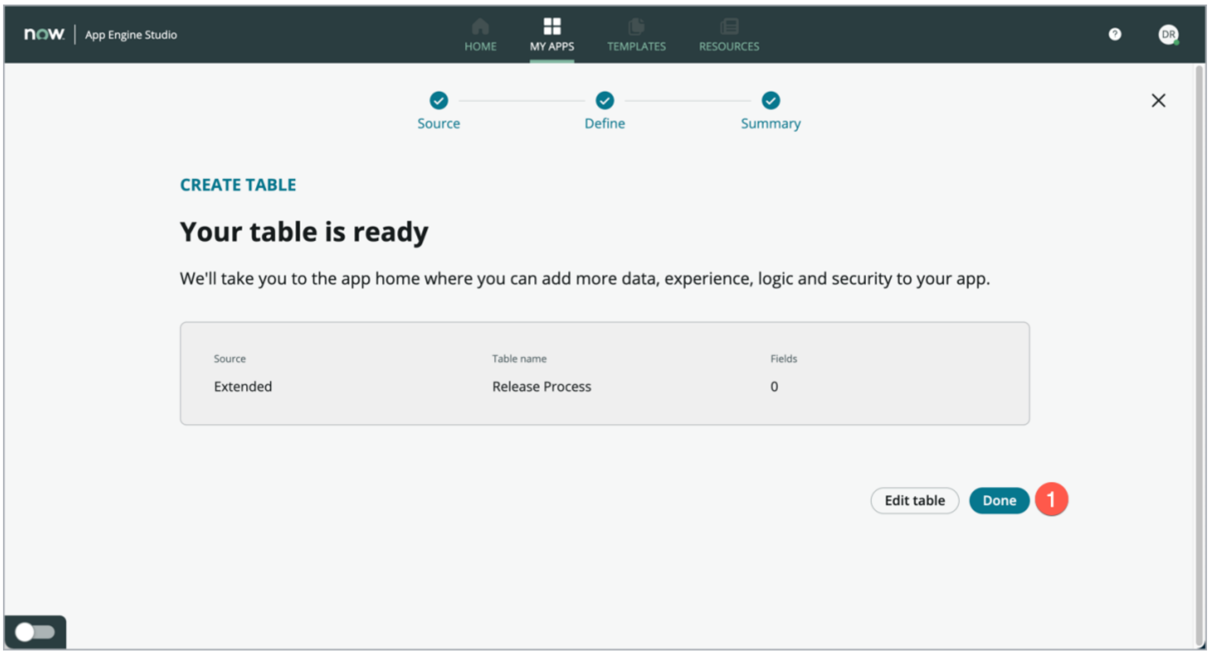
Click “**Continue**” (3) to move on.



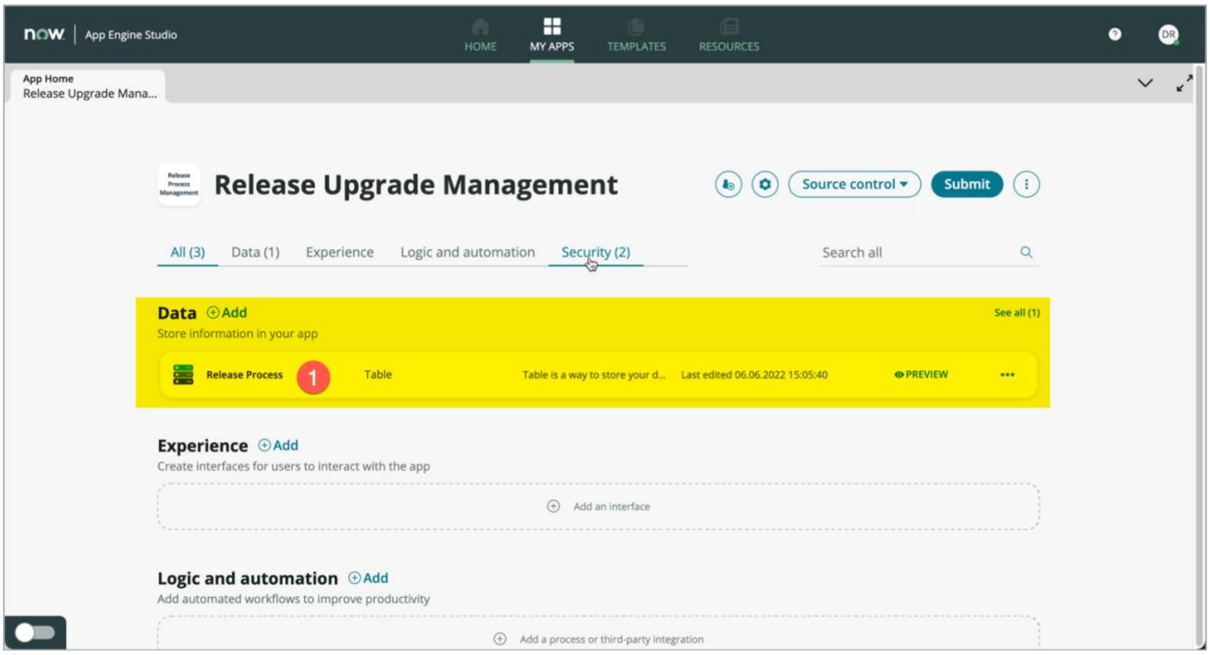
App Engine Studio will now create your table (extended from the “Task” table).



On the next screen, which appears after a moment, click "**Done**" (1).



Back on the Dashboard of your App, you can find the newly created "**Release Process**" (1) table below the "**Data**" section – see screenshot below:

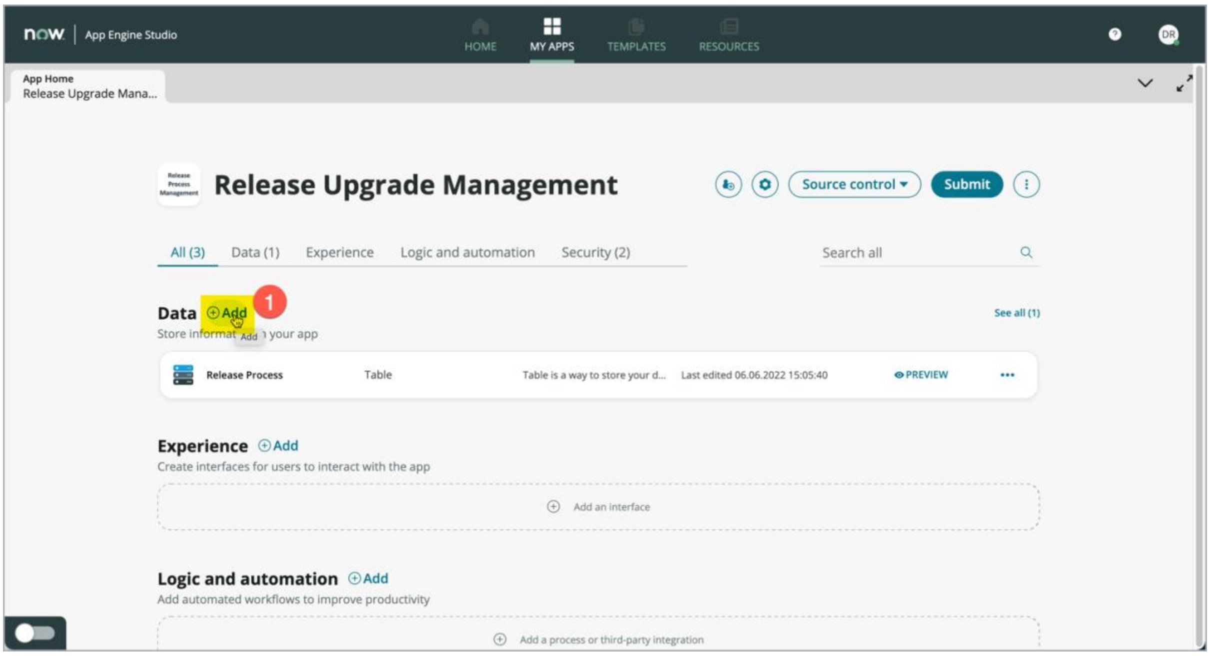


Great, you created the first table "Release Process"!

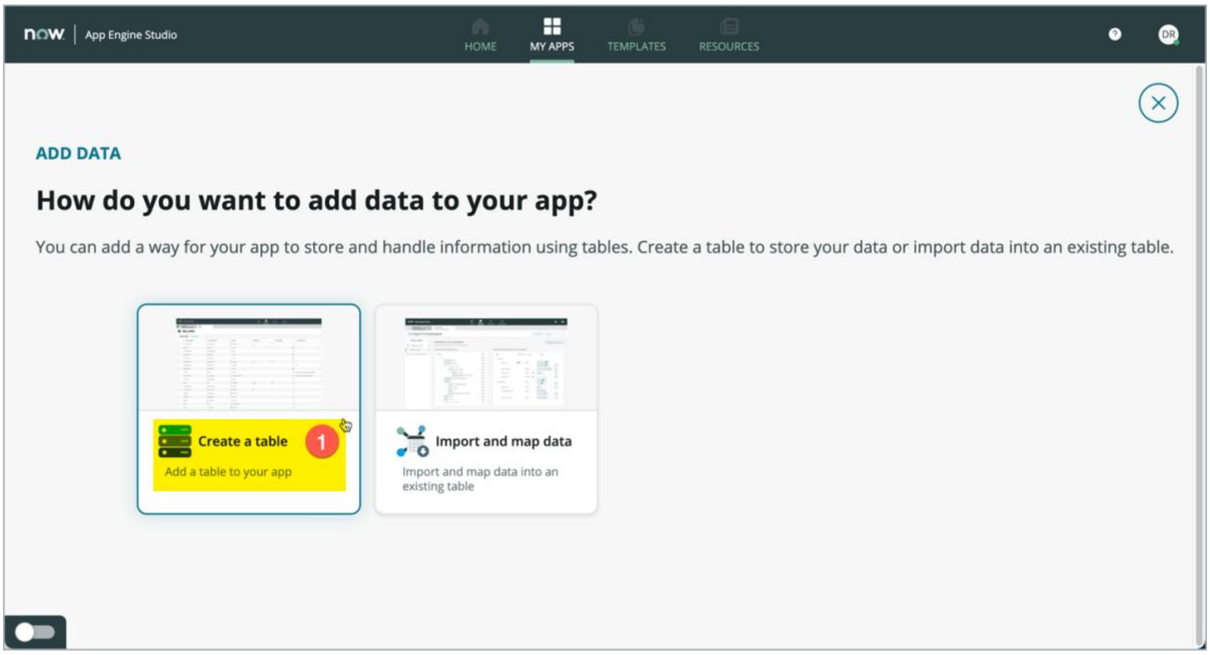
2.1.2 Create the “Release Process Task” table

Now, let’s do that once more and create the “**Release Process Task**” table, which will also be extended from the Task table.

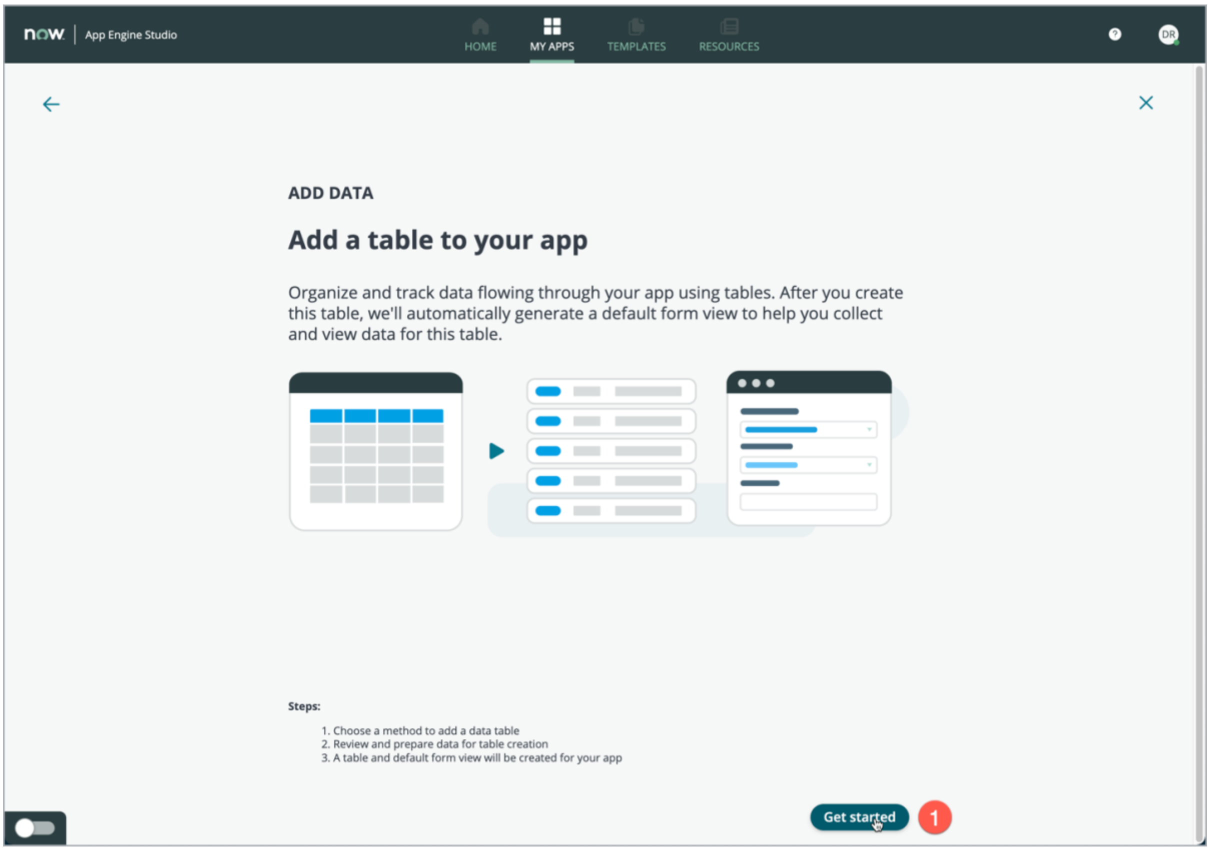
From the Dashboard of your Application, click “Add” (1) in the “Data” section, to start the table creation wizard for our second table – the “Release Process Task” table.



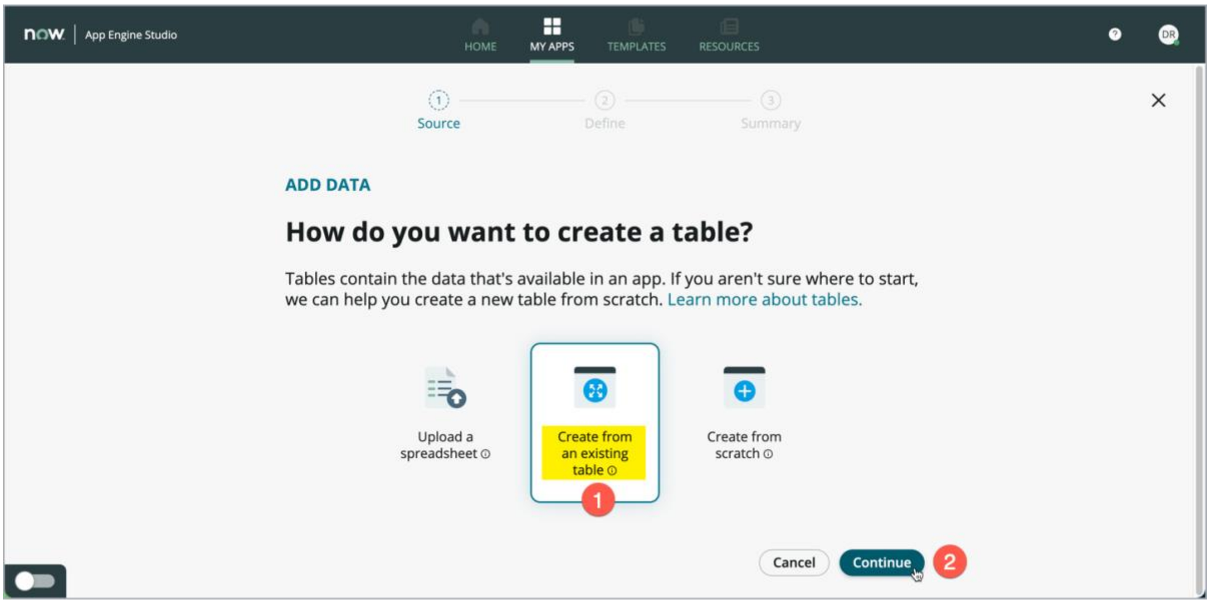
Again, click “**Create a table**”



In the next screen, click “**Get started**”

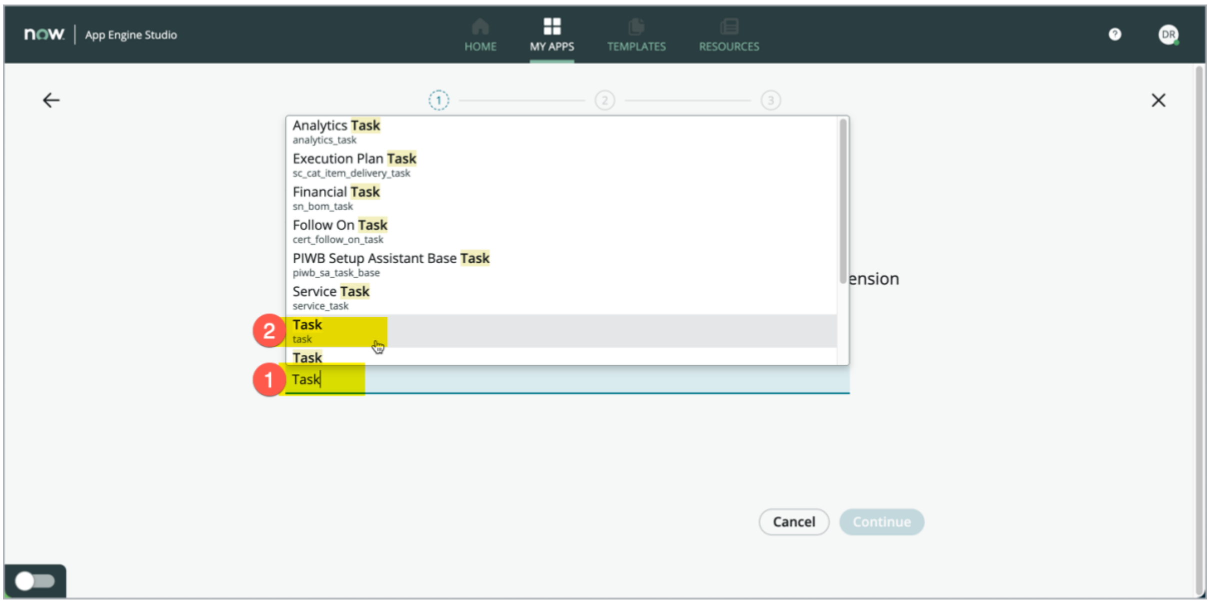


Like for the first table, click “**Create from an existing table**” (1) and click “**Continue**” (2).

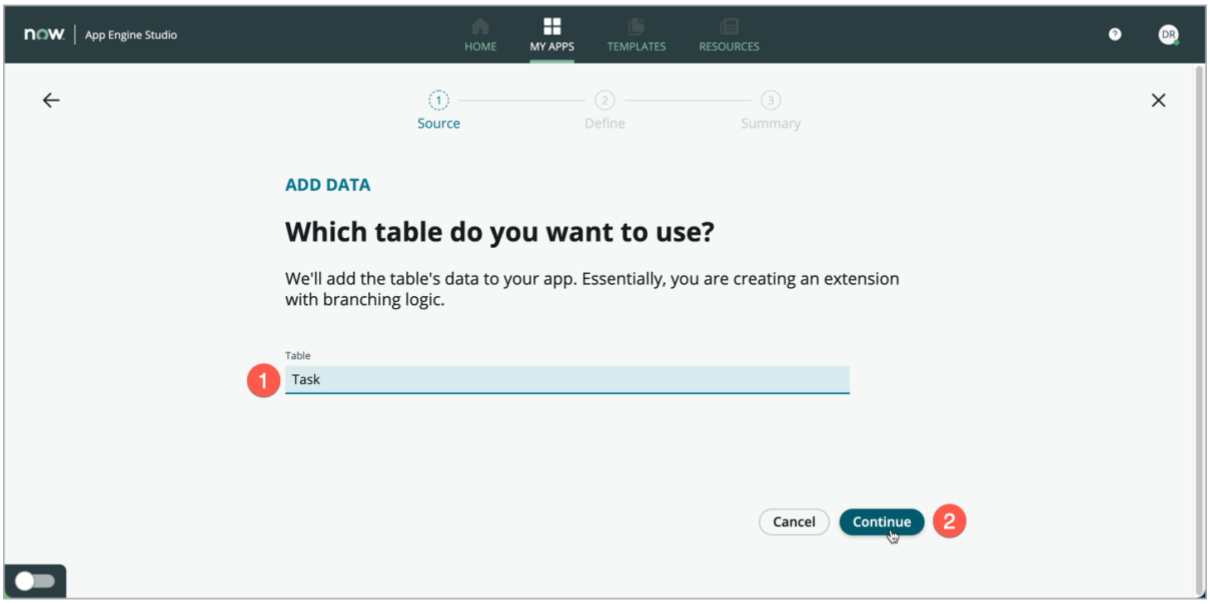


This table is also extended from Task (see screenshot below).

Note:
Do not expect to extend ALL tables from "Task" in the future. The "Task" tables serve as a "first class citizen", if you want to have tables that stored data, which relates 1:1 to task based data, like a "Process Task" (or the overall "Release Process", which is a huge Task effectively).



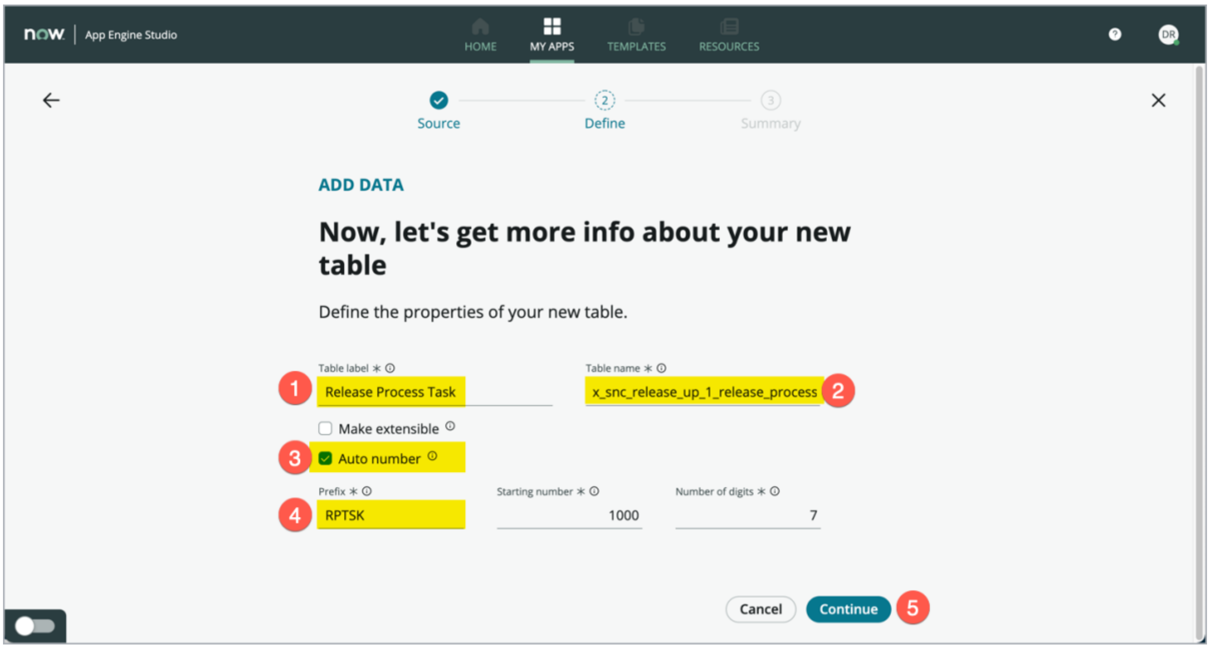
Select the "**Task**" (1) table (like shown in the screenshot) and click "**Continue**" (2).



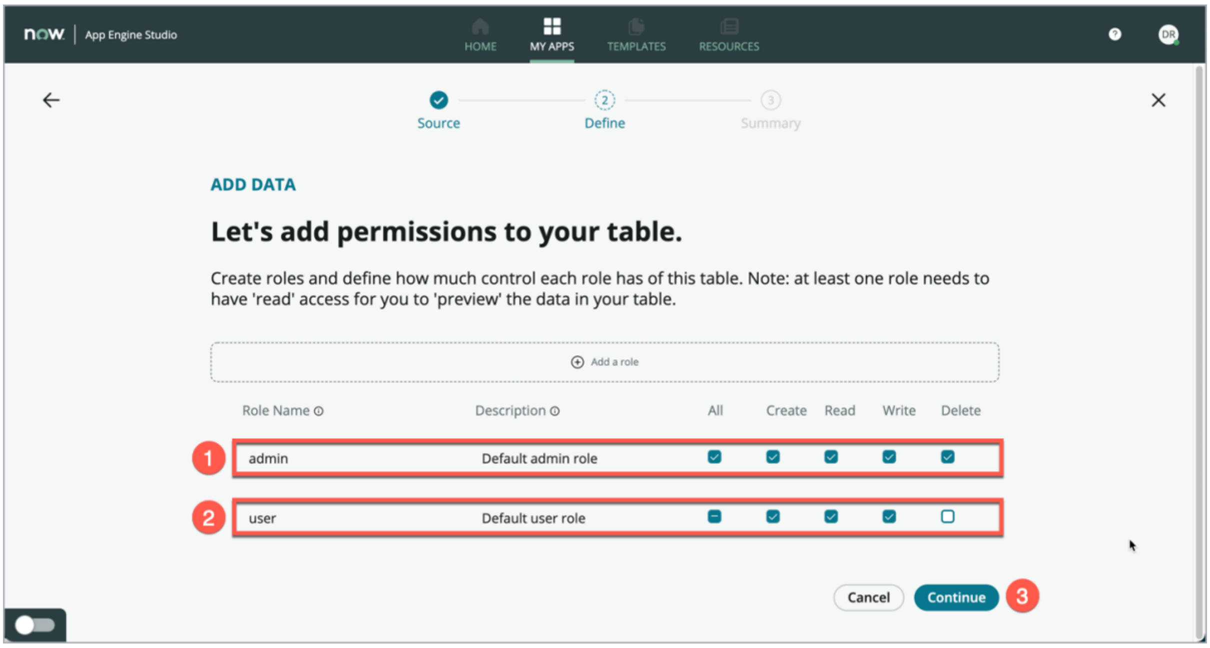
Enter the "Table label" (1) as "**Release Process Task**" and do not change the "Table name" (2) field, which will be populated automatically.

Mark the "**Auto number**" (3) field and enter a valid "**Prefix**" (use "**RPTSK**" (4) here).

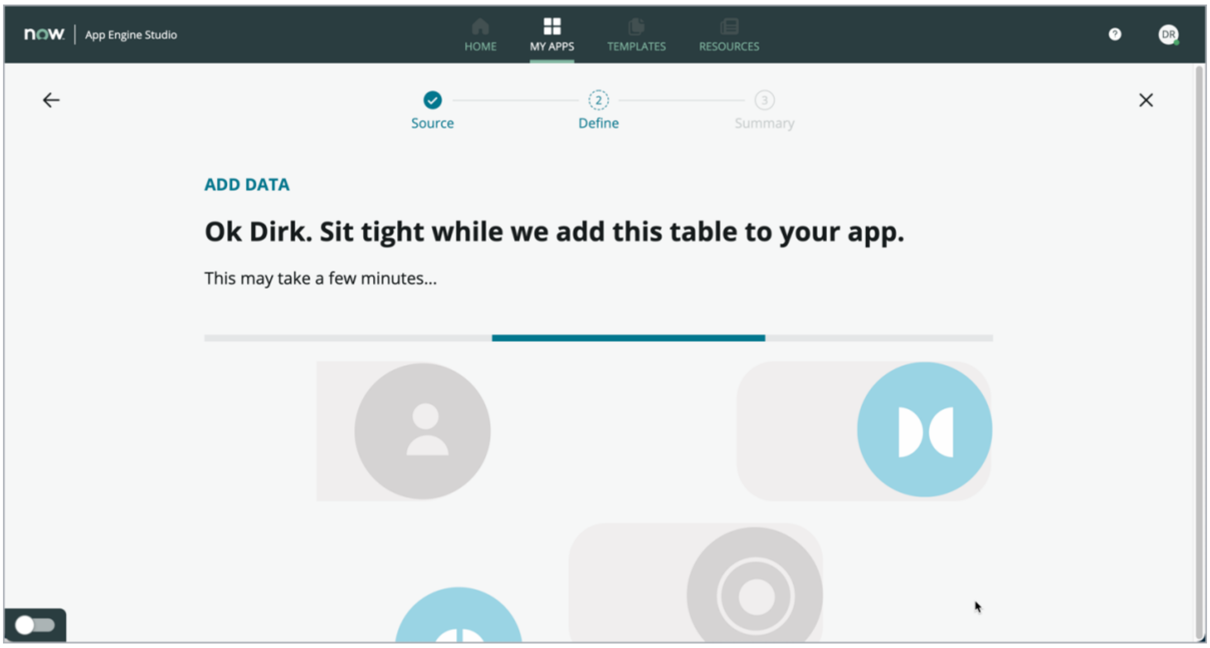
Click "**Continue**" (5) when done to move on.



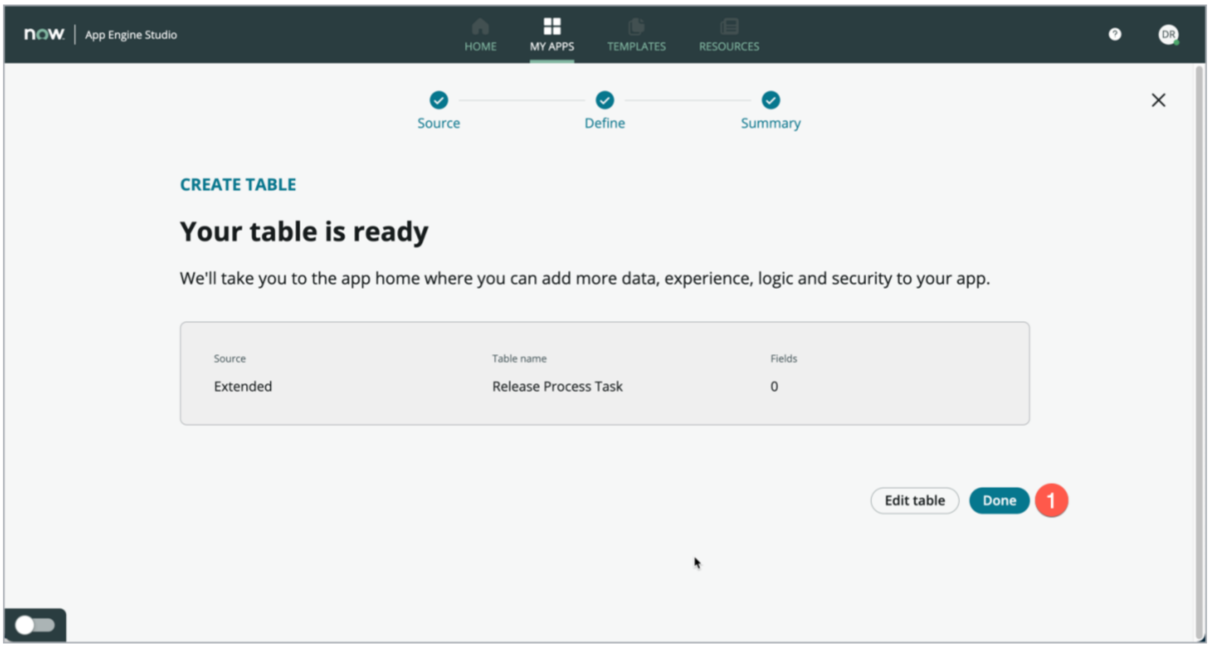
Now, set the permissions for the Roles of the App (1) & (2), just like you did for the first table (see screenshot below) and hit "Continue" (3).



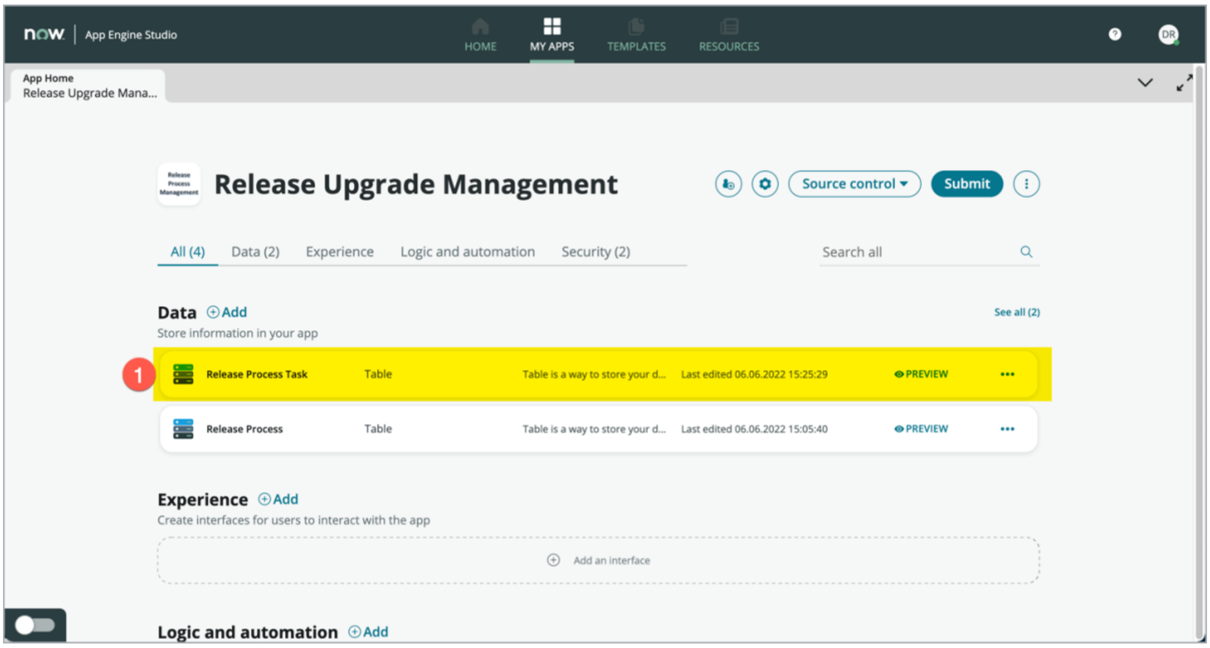
ServiceNow will now create the table:



After the wizard finishes, click “**Done**” (1).



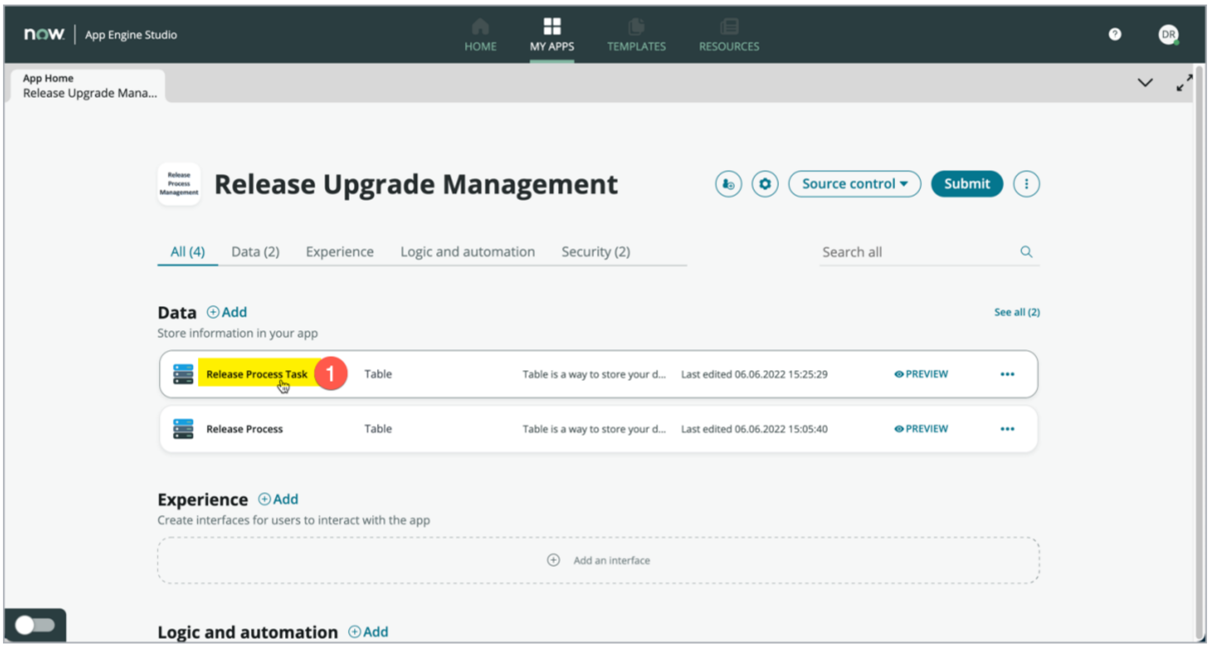
Find your newly created table now also under the "Data" section on the Application Dashboard:



Add more fields to the table

Now, that the table is created, let's add one more field to the table.

Open Table Builder by clicking the "**Release Process Task**" table on its Name (see screenshot below):



Note:
Alternatively, you can click the three dots (1) on the right side of the table row and select "Edit" (2) - see screenshot below.

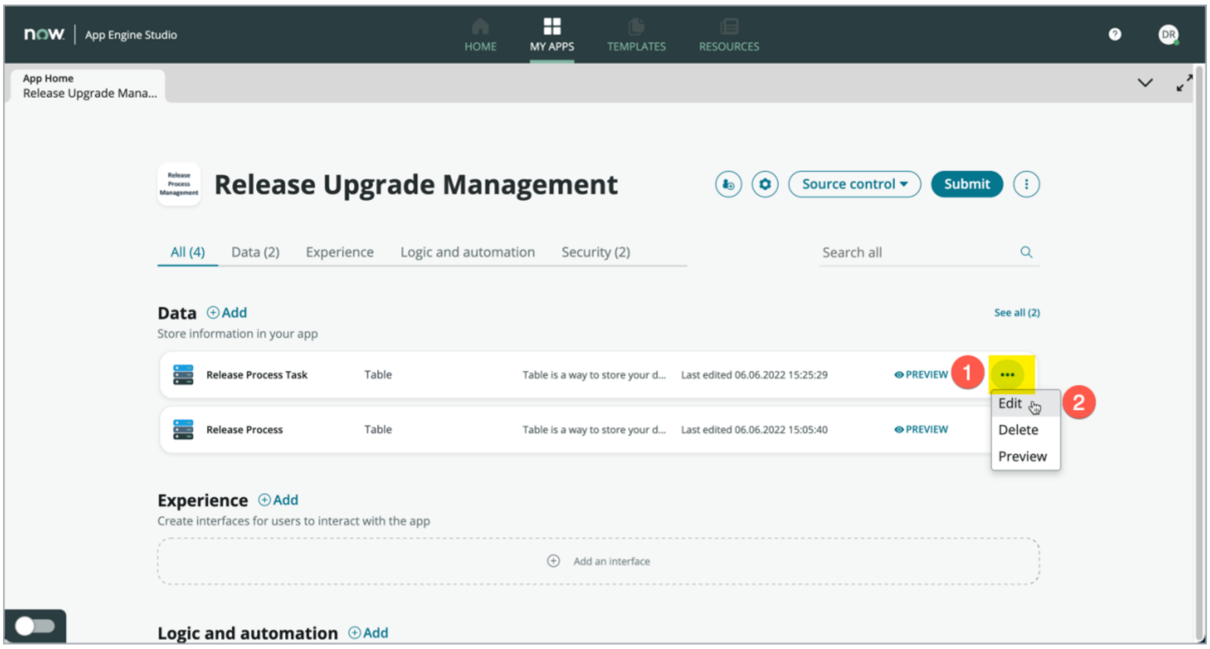
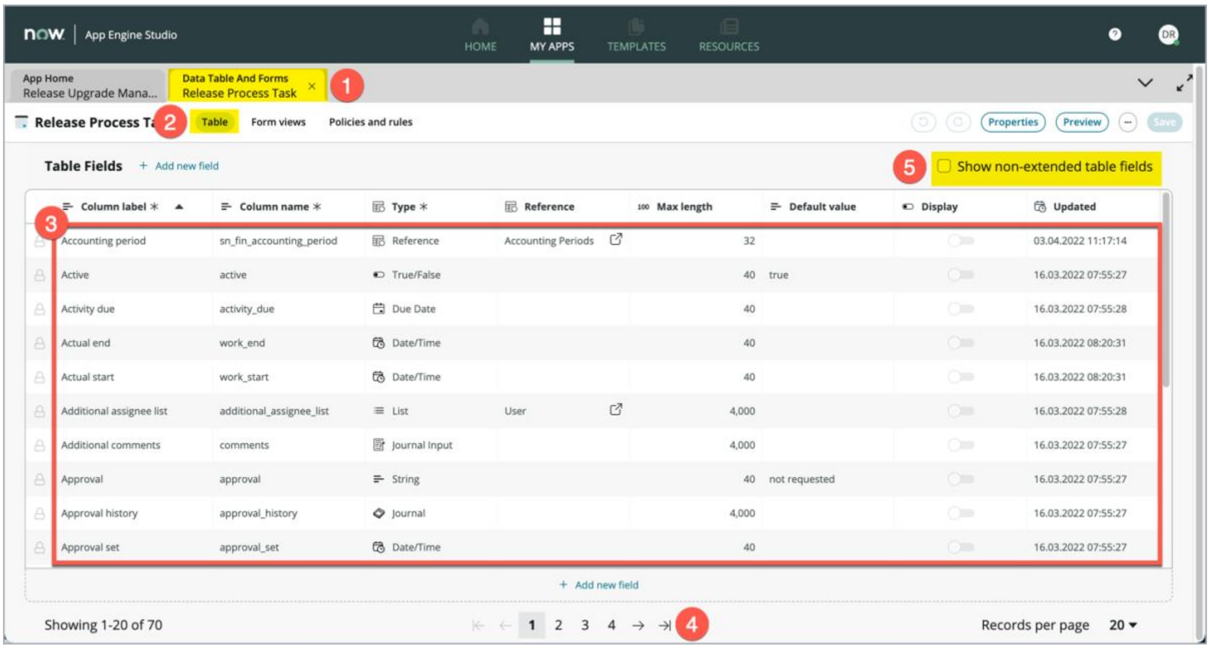


Table Builder opens on a new tab (1) in App Engine Studio. You can see that you are currently viewing the “Table” view (2). The main area of the screen (3) shows the fields of the table. As you can see, there are already several fields (in fact 70 fields – see bottom-left in the screenshot) in the table. Those are the fields which are “inherited” from the extended table “Task” (which we chose during table creation).

You cannot remove any of those inherited fields from your table here. But you can add your own fields to those already existing.

Selecting the Checkbox “Show non-extended table field” (5), you can filter the list of fields to those, which are added to this table AFTER extending from Task.

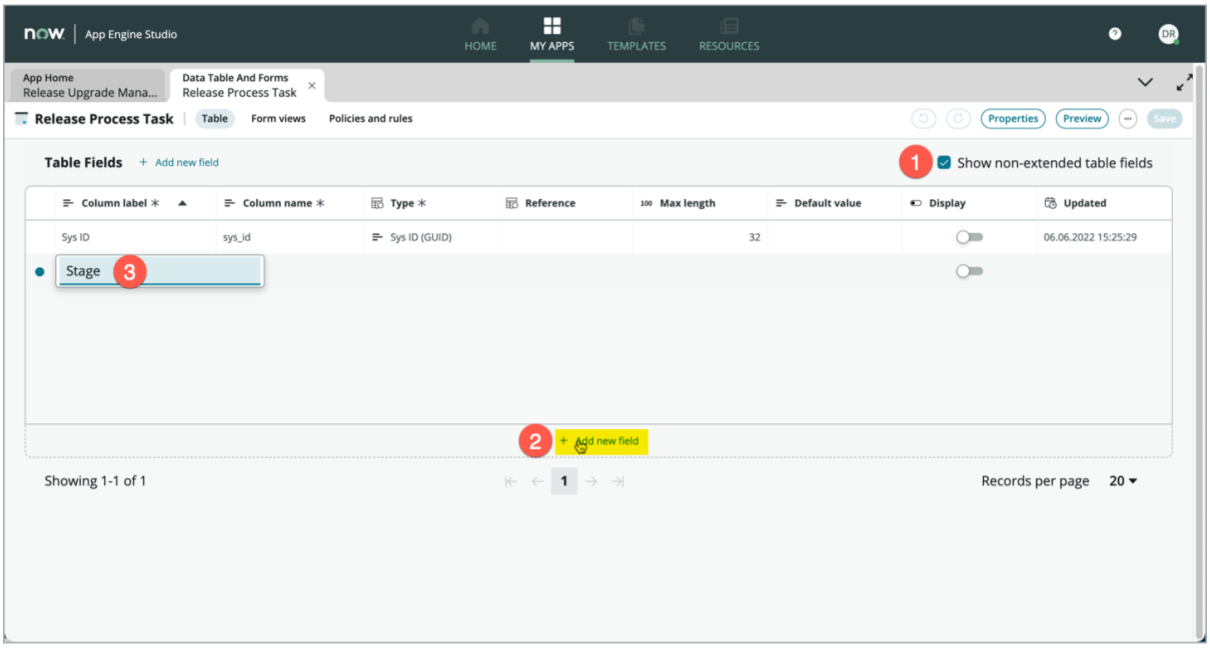
Mark the Checkbox (5) and move to the instructions below.



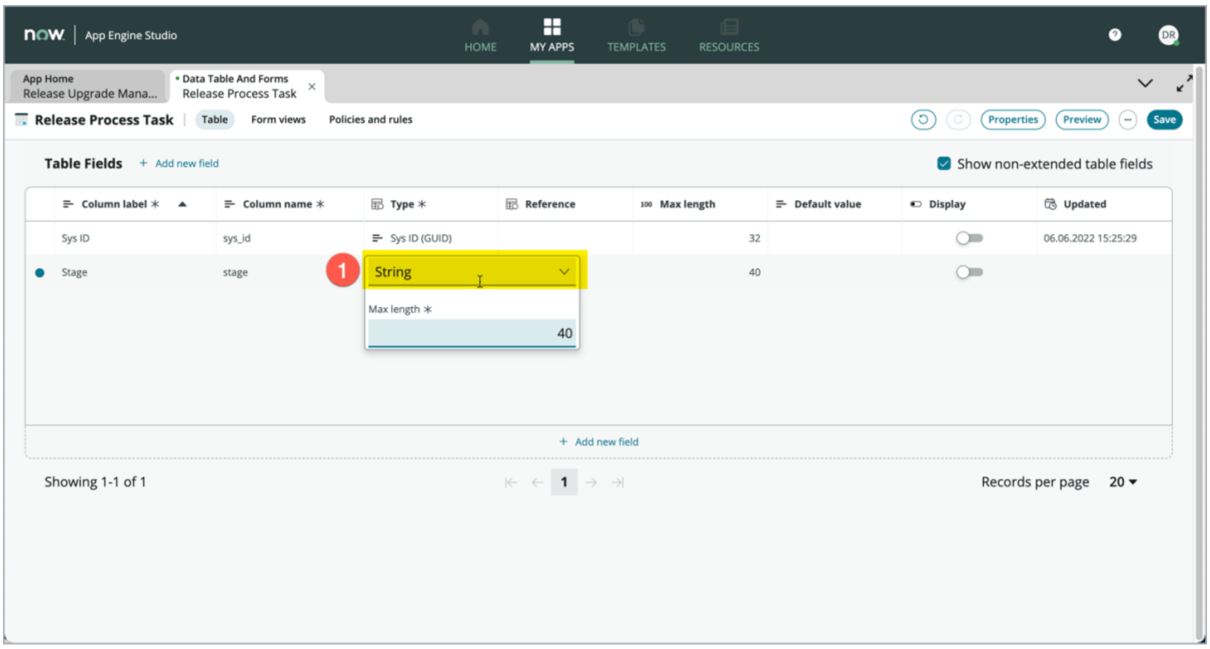
In the next screenshot, you can see, that I marked the Checkbox (1), so that the only field left is the field “Sys ID”.

Info:
The Sys ID is a System field, which occurs in EVERY table in ServiceNow and holds a unique ID (a GUID) to identify each record in the table. ServiceNow adds this field for you, and you cannot delete it.

To add a new field to your table, click “**Add new field**” (2) and enter the “**Column label**” (3) in the field like shows in the screenshot below.

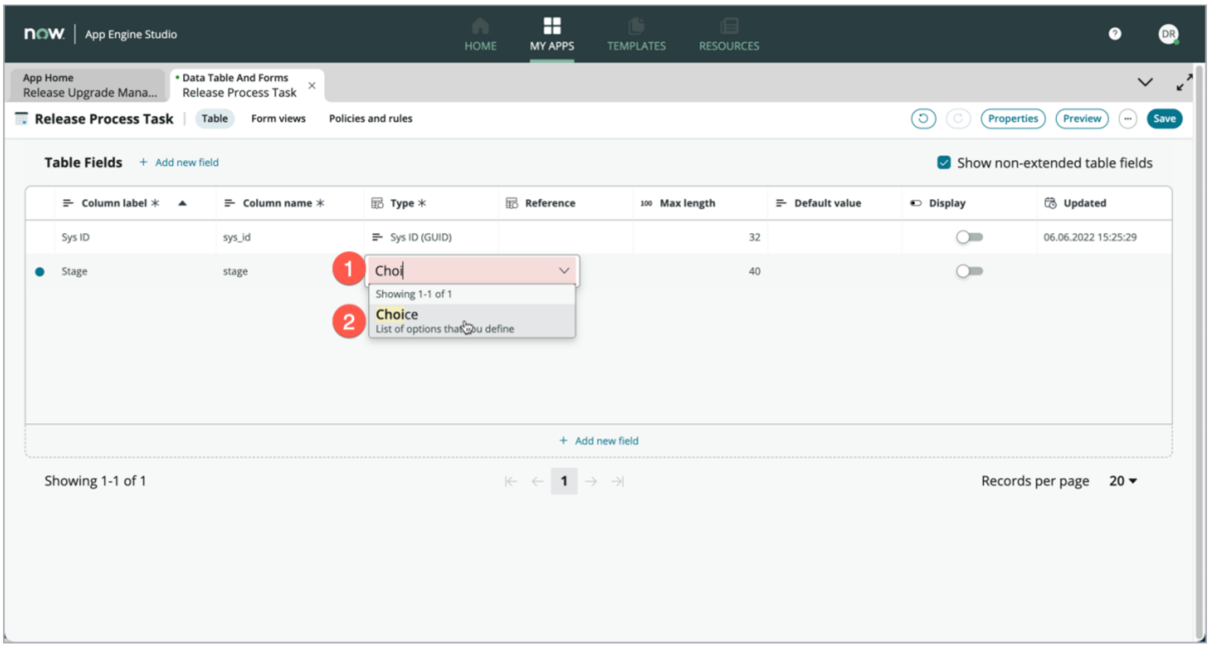


Hit the **<ENTER>** Key and click the row in the “**Type**” column

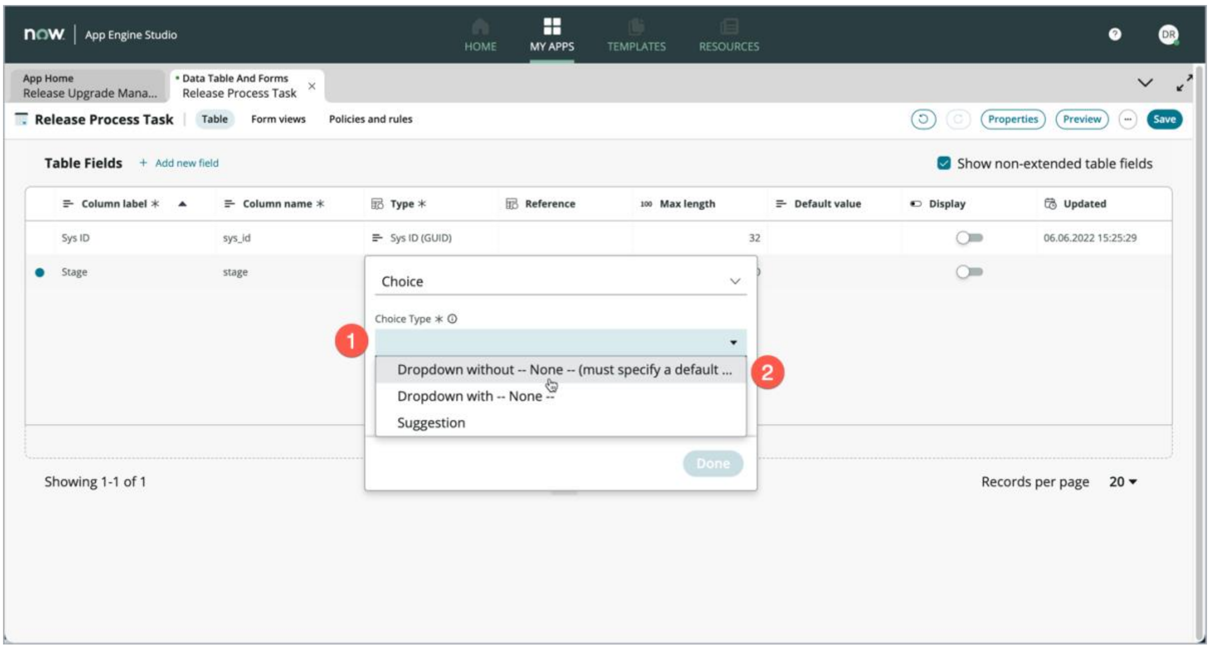


That opens a dropdown field, where you can select the field type (1) of your new column in your table. The field type defaults to “String”.

Change the Field to “**Choice**” by first typing “Choi” (1) in the upper field, and the selecting the “Choice” (2) entry from the dropdown that appears.

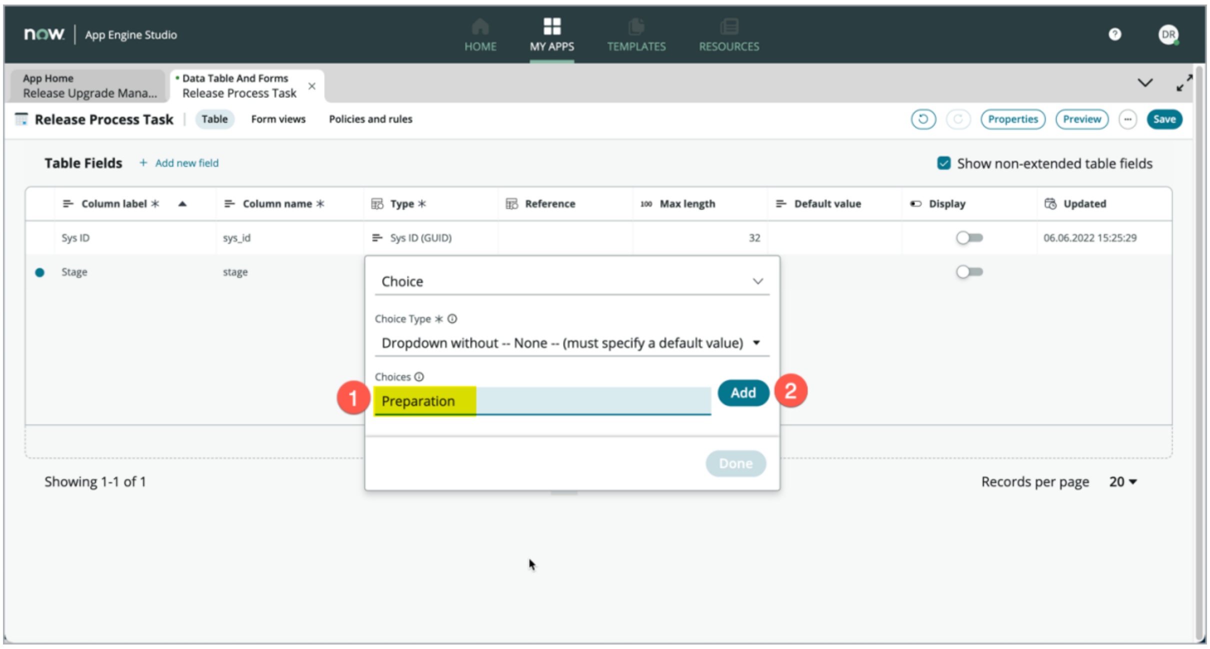


In the “**Choice Type**” field (1) that appears, select the first option “**Dropdown without – None -- ….**” (2).



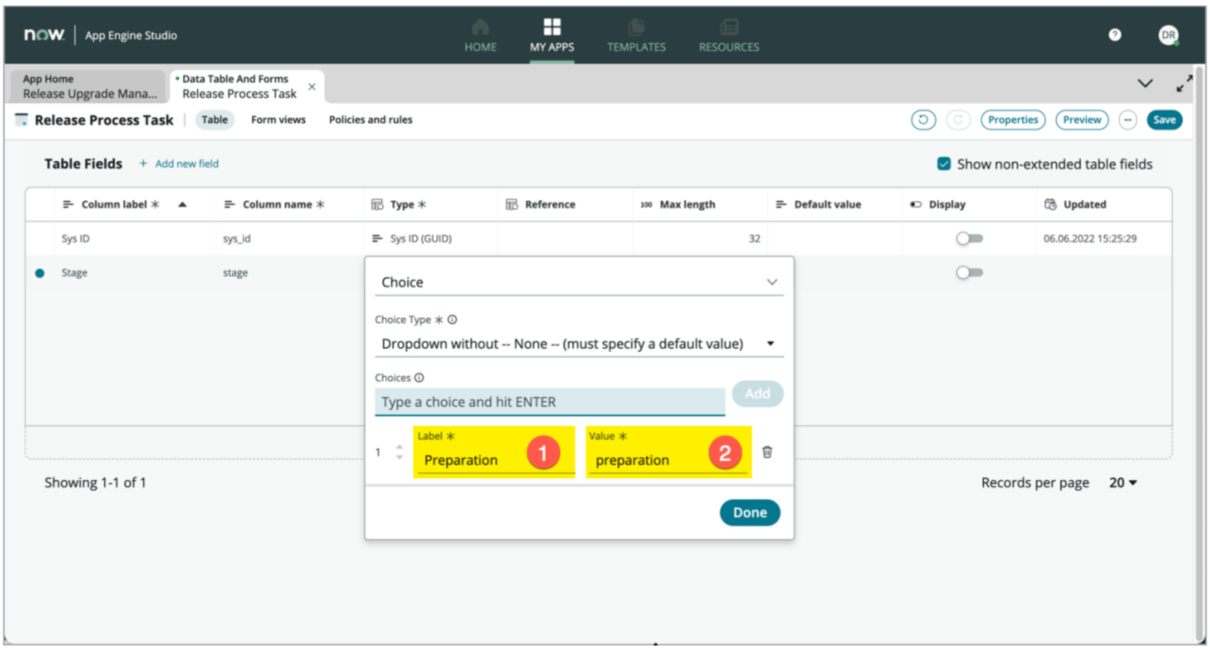
Remember / Review:
From our Excel file in the first Column, we found a total of four Stages of the process (“**Preparation**”, “**Execution**”, “**Tests**” and “**Closing**”).

Let’s use those values to create Choices.
Enter “**Preparation**” into the “Choices” (1) field and click “**Add**” (2).



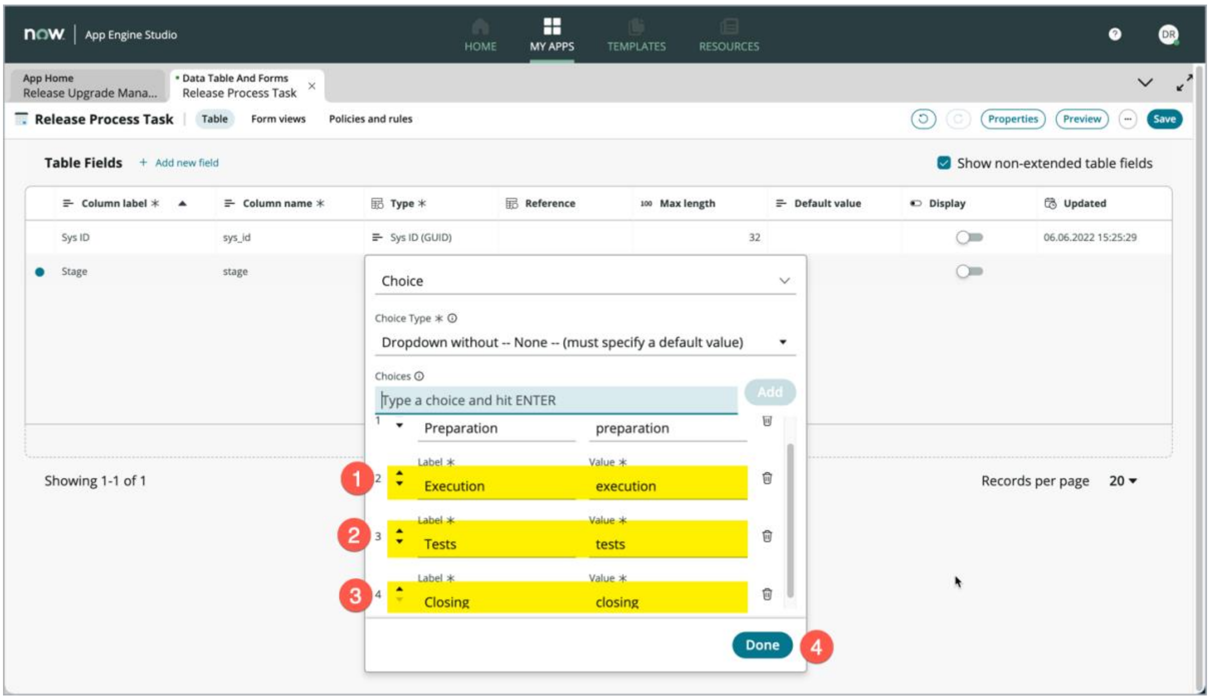
Note, that in the list below the “**Choices**” field, your entry is listed. There are two columns, because **for each choice**, there is a “**Label**” (1) (which is used to show to the end-user on the screen) and a “**Value**” (2) which will be stored in the database field for this choice selection.

Note:
The “**Value**” is written **ALL in lowercase**. This is important to review and set up accordingly. Mention, that the value of the choice in the database must be unique per choice field, and that the values are case-sensitive.

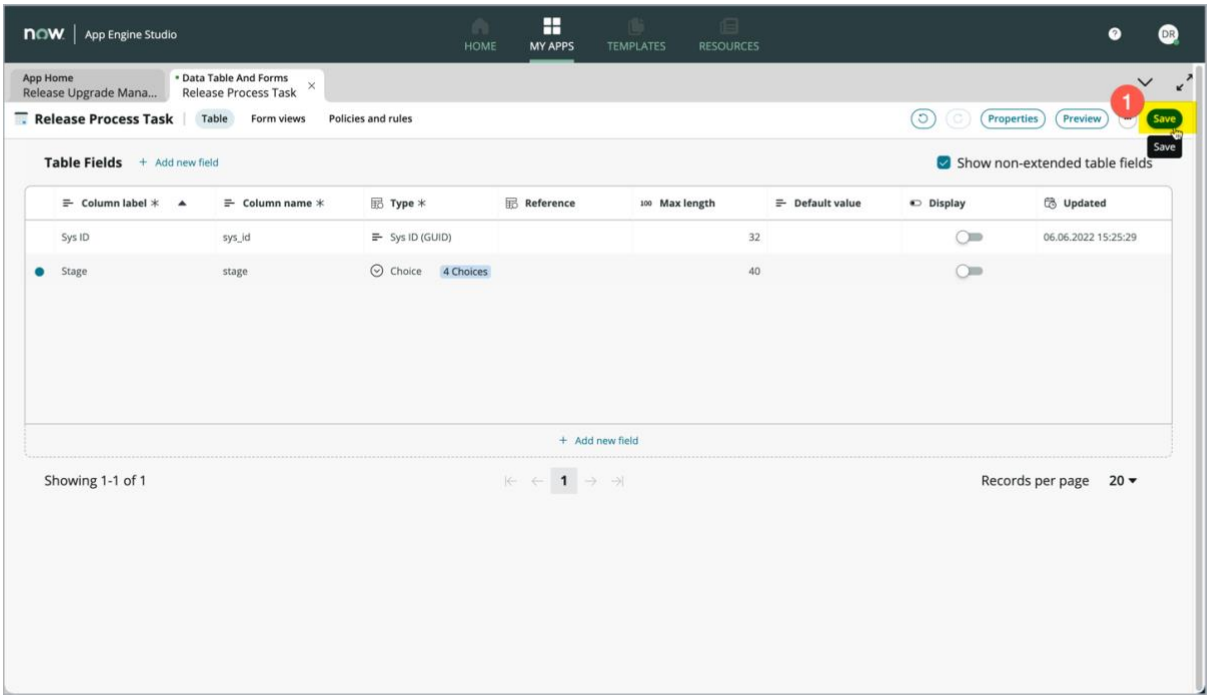


In the same way, enter the Choice Values for “**Execution**” (1), “**Tests**” (2) and “**Closing**” (3). Again **MAKE SURE**, that the values all are done in **lowercase**.

Click “**Done**” (4) to close the popup.



Click “**Save**” (1) to make the changes permanent



Congratulations, you just have created another table with an additional field.

2.1.3 Create the “Release Task Blueprint” table

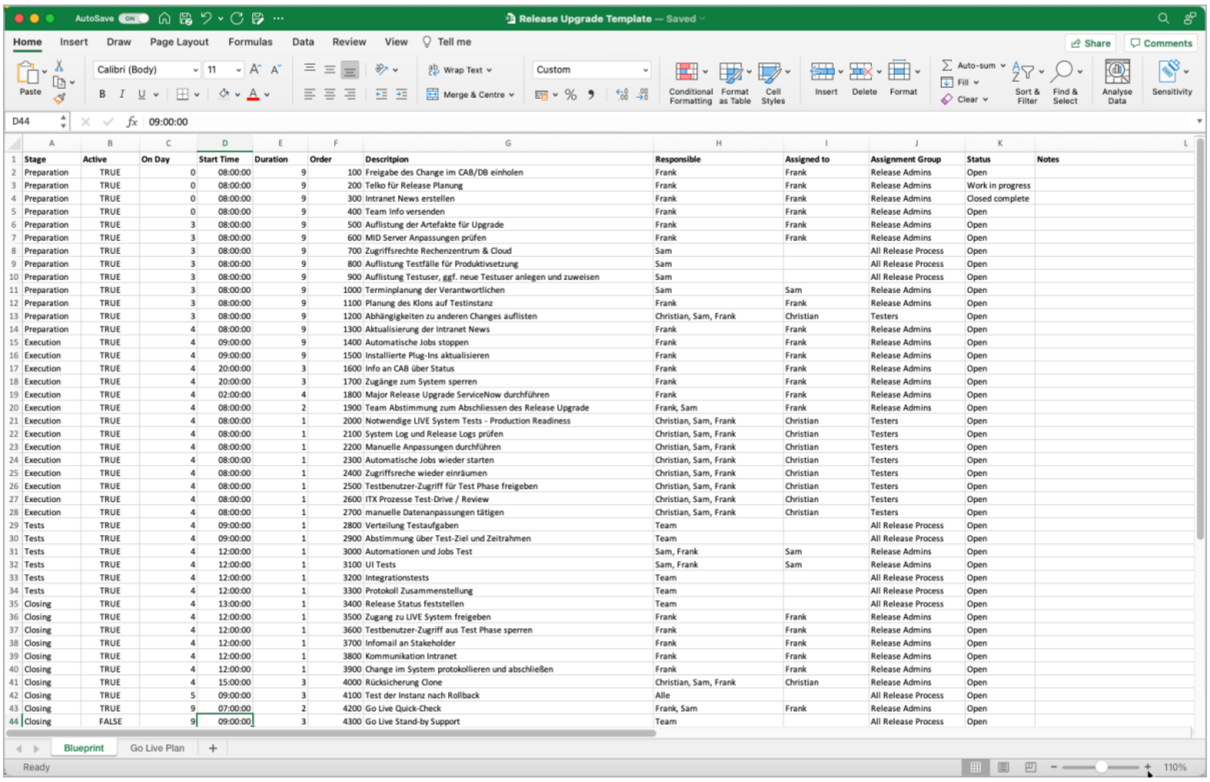
For our App, **we started from an Excel spreadsheet**, where we can find data, that until today, was used to drive the process. (review the screenshot of the spreadsheet above in this document).

Remember, that in **today’s process**, **one Excel file** (or at least the “Go Live Plan” Tab) always related to **ONE** Release Upgrade **Process**.

We now want to leverage the list of available Tasks as a **template/blueprint** for new Processes to be created.

Note:
Before we import the Excel file to our App, we want to “clean-up” this file a bit, and modify the columns to match our needs in the App.

Let’s have a look at the modified version of the Excel file from above:



Stage	Active	On Day	Start Time	Duration	Order	Descritpion	Responsible	Assigned to	Assignment Group	Status	Notes
Preparation	TRUE	0	08:00:00	9	100	Freigabe des Change im CAB/DB einholen	Frank	Frank	Release Admins	Open	
Preparation	TRUE	0	08:00:00	9	200	Telko für Release Planung	Frank	Frank	Release Admins	Work in progress	
Preparation	TRUE	0	08:00:00	9	300	Intranet News erstellen	Frank	Frank	Release Admins	Closed complete	
Preparation	TRUE	0	08:00:00	9	400	Team Info versenden	Frank	Frank	Release Admins	Open	
Preparation	TRUE	3	08:00:00	9	500	Auflistung der Artefakte für Upgrade	Frank	Frank	Release Admins	Open	
Preparation	TRUE	3	08:00:00	9	600	MID Server Anpassungen prüfen	Frank	Frank	Release Admins	Open	
Preparation	TRUE	3	08:00:00	9	700	Zugriffsrechte Rechenzentrum & Cloud	Sam		All Release Process	Open	
Preparation	TRUE	3	08:00:00	9	800	Auflistung Testfälle für Produktivsetzung	Sam		All Release Process	Open	
Preparation	TRUE	3	08:00:00	9	900	Auflistung Testuser, ggf. neue Testuser anlegen und zuweisen	Sam		All Release Process	Open	
Preparation	TRUE	3	08:00:00	9	1000	Terminplanung der Verantwortlichen	Sam	Sam	Release Admins	Open	
Preparation	TRUE	3	08:00:00	9	1100	Planung des Klons auf Testinstanz	Frank	Frank	Release Admins	Open	
Preparation	TRUE	3	08:00:00	9	1200	Abhängigkeiten zu anderen Changes auflisten	Christian, Sam, Frank	Christian	Testers	Open	
Preparation	TRUE	4	08:00:00	9	1300	Aktualisierung der Intranet News	Frank	Frank	Release Admins	Open	
Execution	TRUE	4	09:00:00	9	1400	Automatische Jobs stoppen	Frank	Frank	Release Admins	Open	
Execution	TRUE	4	09:00:00	9	1500	Installierte Plug-Ins aktualisieren	Frank	Frank	Release Admins	Open	
Execution	TRUE	4	20:00:00	3	1600	Info an CAB über Status	Frank	Frank	Release Admins	Open	
Execution	TRUE	4	20:00:00	3	1700	Zugänge zum System sperren	Frank	Frank	Release Admins	Open	
Execution	TRUE	4	02:00:00	4	1800	Major Release Upgrade ServiceNow durchführen	Frank	Frank	Release Admins	Open	
Execution	TRUE	4	08:00:00	2	1900	Team Abstimmung zum Abschliessen des Release Upgrade	Frank, Sam	Frank	Release Admins	Open	
Execution	TRUE	4	08:00:00	1	2000	Notwendige LIVE System Tests - Production Readiness	Christian, Sam, Frank	Christian	Testers	Open	
Execution	TRUE	4	08:00:00	1	2100	System Log und Release Logs prüfen	Christian, Sam, Frank	Christian	Testers	Open	
Execution	TRUE	4	08:00:00	1	2200	Manuelle Anpassungen durchführen	Christian, Sam, Frank	Christian	Testers	Open	
Execution	TRUE	4	08:00:00	1	2300	Automatische Jobs wieder starten	Christian, Sam, Frank	Christian	Testers	Open	
Execution	TRUE	4	08:00:00	1	2400	Zugriffsreche wieder einräumen	Christian, Sam, Frank	Christian	Testers	Open	
Execution	TRUE	4	08:00:00	1	2500	Testbenutzer-Zugriff für Test Phase freigeben	Christian, Sam, Frank	Christian	Testers	Open	
Execution	TRUE	4	08:00:00	1	2600	ITX Prozesse Test-Drive / Review	Christian, Sam, Frank	Christian	Testers	Open	
Execution	TRUE	4	08:00:00	1	2700	manuelle Datenanpassungen tätigen	Christian, Sam, Frank	Christian	Testers	Open	
Tests	TRUE	4	09:00:00	1	2800	Verteilung Testaufgaben	Team		All Release Process	Open	
Tests	TRUE	4	09:00:00	1	2900	Abstimmung über Test-Ziel und Zeitrahmen	Team		All Release Process	Open	
Tests	TRUE	4	12:00:00	1	3000	Automationen und Jobs Test	Sam, Frank	Sam	Release Admins	Open	
Tests	TRUE	4	12:00:00	1	3100	UI Tests	Sam, Frank	Sam	Release Admins	Open	
Tests	TRUE	4	12:00:00	1	3200	Integrationstests	Team		All Release Process	Open	
Tests	TRUE	4	12:00:00	1	3300	Protokoll Zusammenstellung	Team		All Release Process	Open	
Closing	TRUE	4	13:00:00	1	3400	Release Status feststellen	Team		All Release Process	Open	
Closing	TRUE	4	12:00:00	1	3500	Zugang zu LIVE System freigeben	Frank	Frank	Release Admins	Open	
Closing	TRUE	4	12:00:00	1	3600	Testbenutzer-Zugriff aus Test Phase sperren	Frank	Frank	Release Admins	Open	
Closing	TRUE	4	12:00:00	1	3700	Infomail an Stakeholder	Frank	Frank	Release Admins	Open	
Closing	TRUE	4	12:00:00	1	3800	Kommunikation Intranet	Frank	Frank	Release Admins	Open	
Closing	TRUE	4	12:00:00	1	3900	Change im System protokollieren und abschließen	Frank	Frank	Release Admins	Open	
Closing	TRUE	4	15:00:00	3	4000	Rücksicherung Clone	Christian, Sam, Frank	Christian	Release Admins	Open	
Closing	TRUE	5	09:00:00	3	4100	Test der Instanz nach Rollback	Alle		All Release Process	Open	
Closing	TRUE	9	07:00:00	2	4200	Go Live Quick-Check	Frank, Sam	Frank	Release Admins	Open	
Closing	FALSE	9	09:00:00	3	4300	Go Live Stand-by Support	Team		All Release Process	Open	

Let’s briefly discuss the meaning of the columns:

- The first column is renamed to “**Stage**” and now holds a value in each row.
    - Note, that you have a “SIMPLE” format in your Excel file before importing. That makes the import much more efficient.

- The “**Active**” Column is intended to define, if each given row is still active and should be used for new Processes to start. If the Row is marked as “Active = FALSE”, it should NOT be used and NOT be copied to new Release Processes.

- The “**On Day**” Column holds the number of days, the task becomes due, relative to the start day of the overall process. This makes the blueprint act “dynamic” for newly created Release Processes.

- The “**Start Time**” field holds the time, which should be used on the specific day for the “Expected start time” of each given task.

- The “**Duration**” field holds an estimated time (in hours), the task will take.

- The “**Description**” is a short description of the task itself to be completed

- The “**Responsible**” column list people involved in that task

- The “**Assigned to**” column hold the ONE person in charge to make sure this task will get completed.

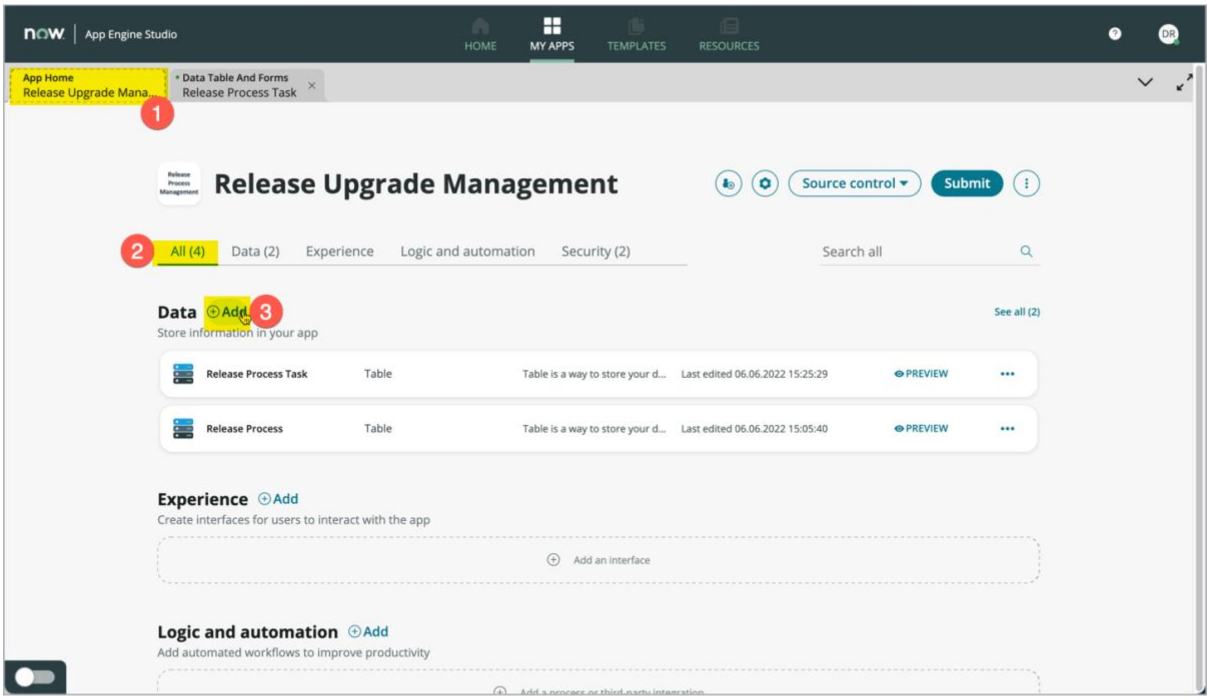
- The “**Assignment Group**” column holds a group of people which is also accountable for this task. E.g. if the “Assigned to” person is not available, another person from the “Assignment Group” may take over the task.

- The “**Status**” column relates to the overall status of that given task

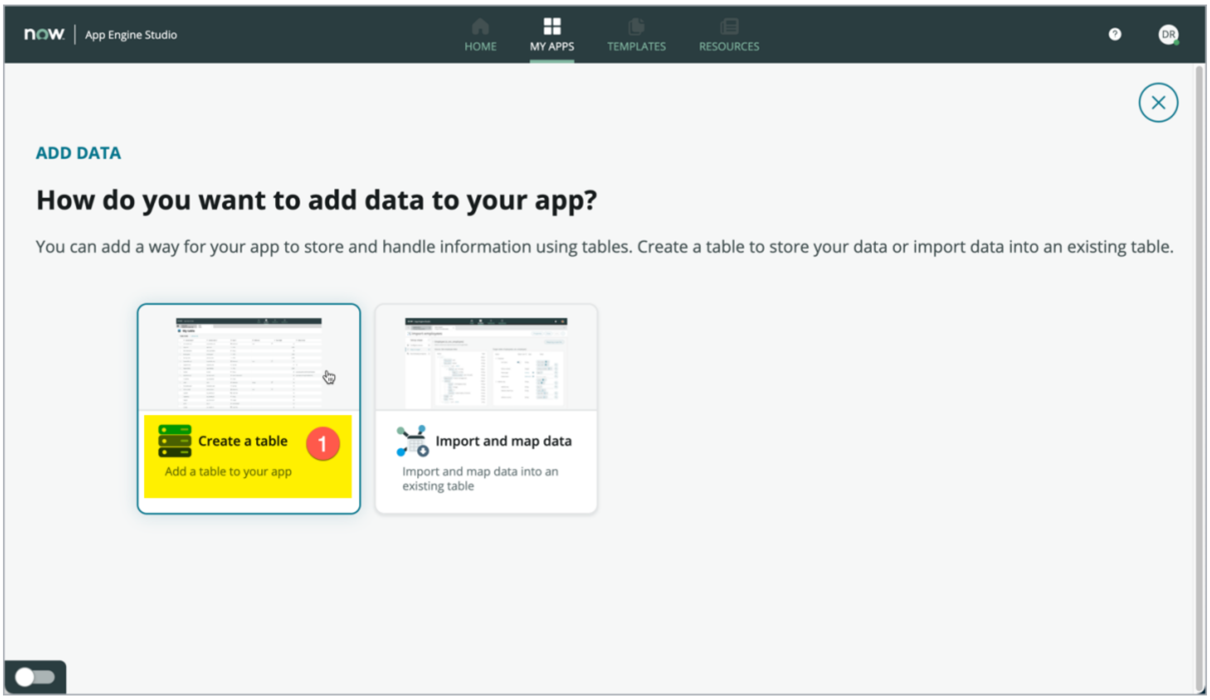
- The “**Notes**” field may hold more details for the task (which can be used as Description in the ServiceNow Task)

Let’s import that Excel file now and create a new database table from it.

Navigate back to your App Dashboard (1) within App Engine Studio and click “**Add**” (3) in the “**Data**” section. Make sure, the “All” tab (2) is selected.

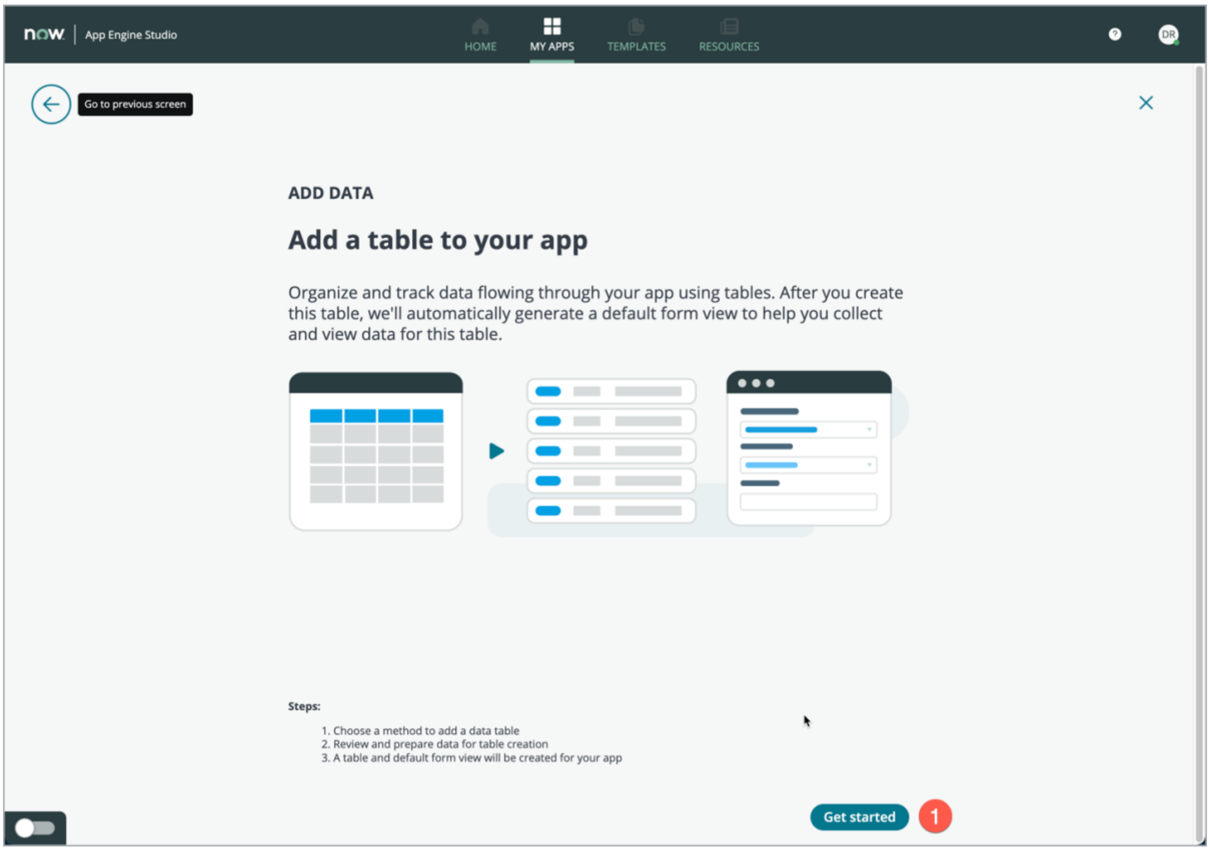


Once more, click “**Create a table**” (1)

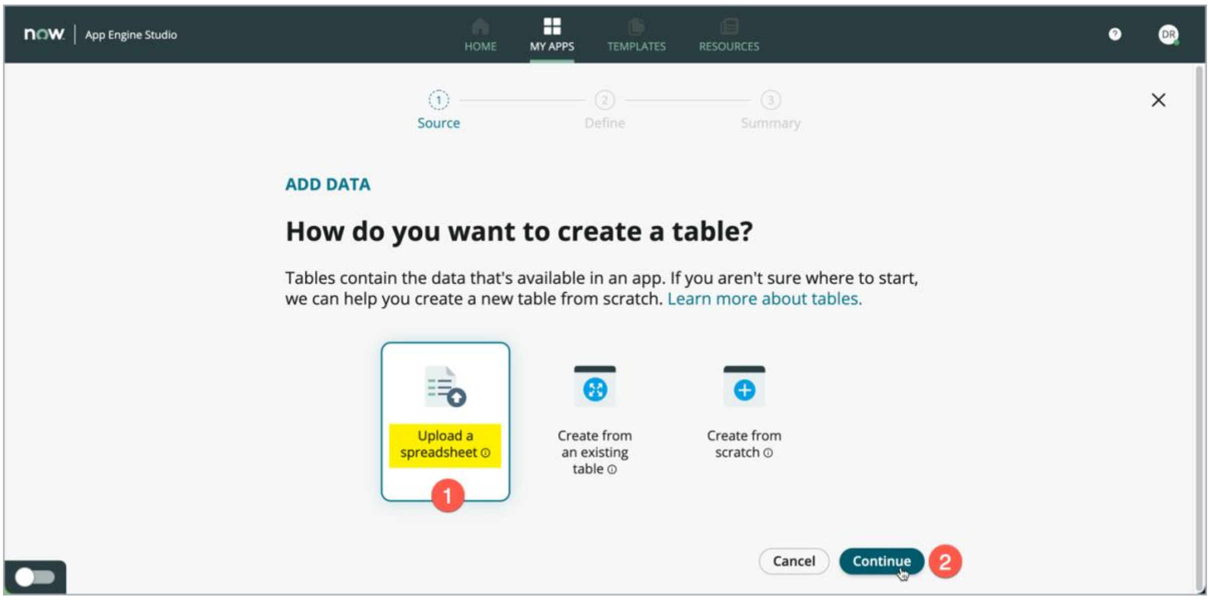


Note:
Even if “**Import and map data**” sounds applicable, that option is used to create imports without creating tables. For our exercise today, we want to create a new table.

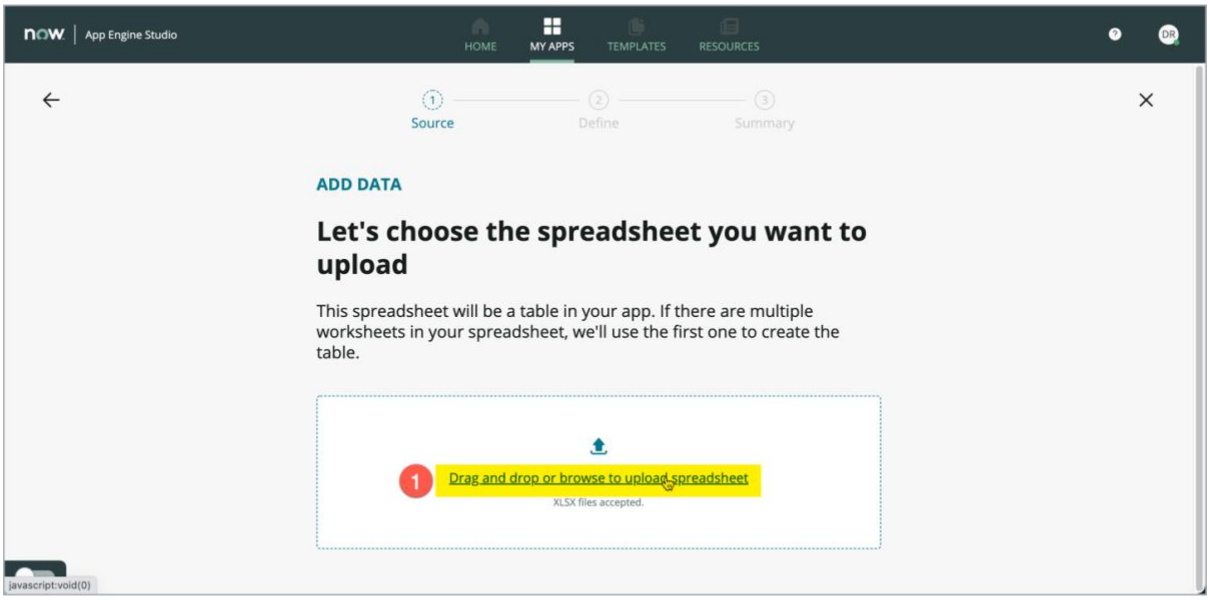
Click “**Get started**” (1). Scroll down a bit, if the button is not visible on your screen.



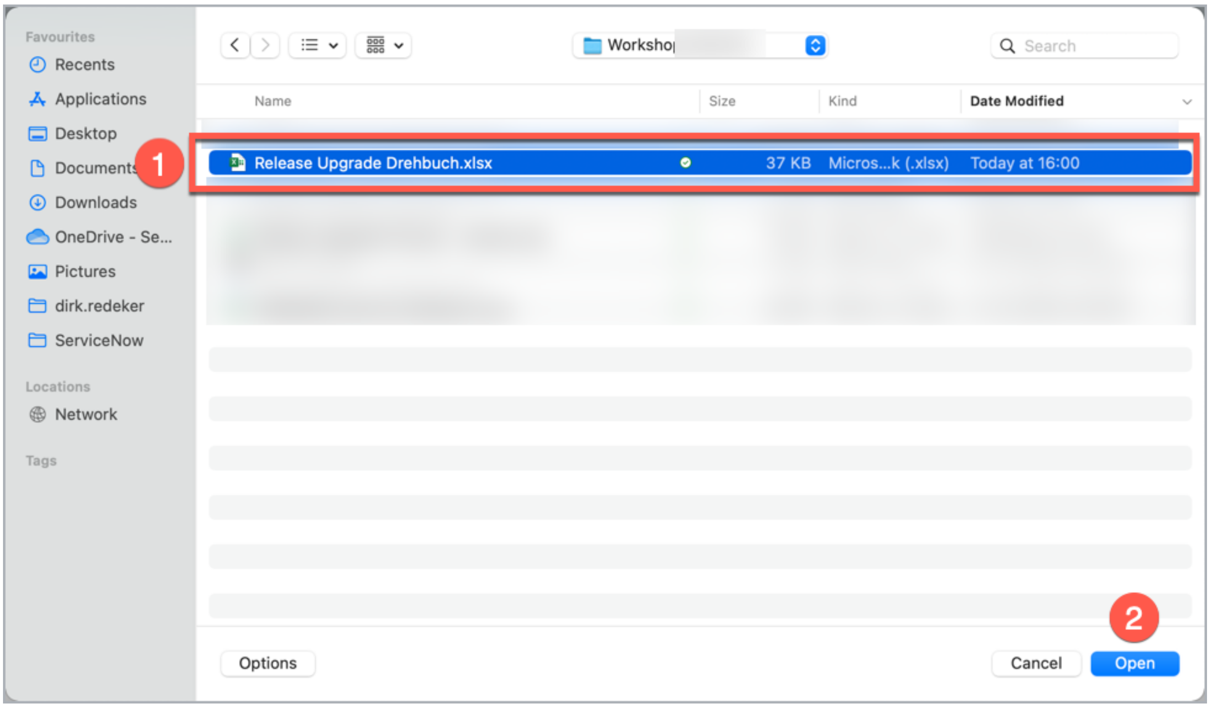
This time, select the left card “**Upload a spreadsheet**” (1) and click “**Continue**” (2)



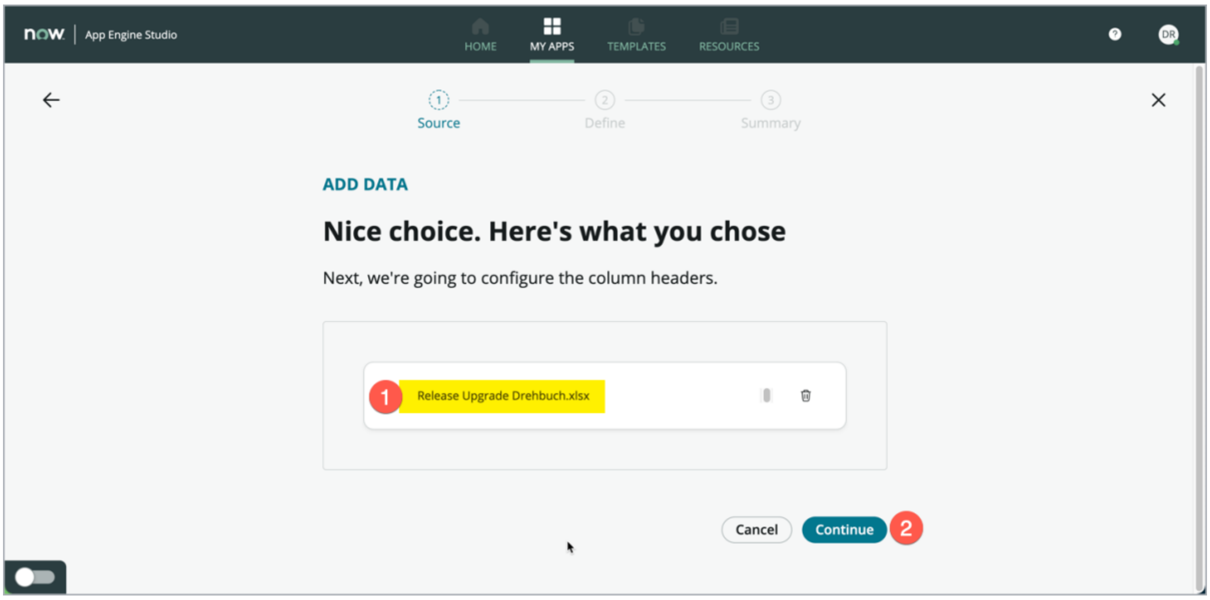
Now, select the Excel file from your local drive, by clicking the Link (1) shown in the screenshot below. This opens the file picker dialog on your local computer.



Find and select the Excel file (1) to be used for the import and click “**Open**” (2).



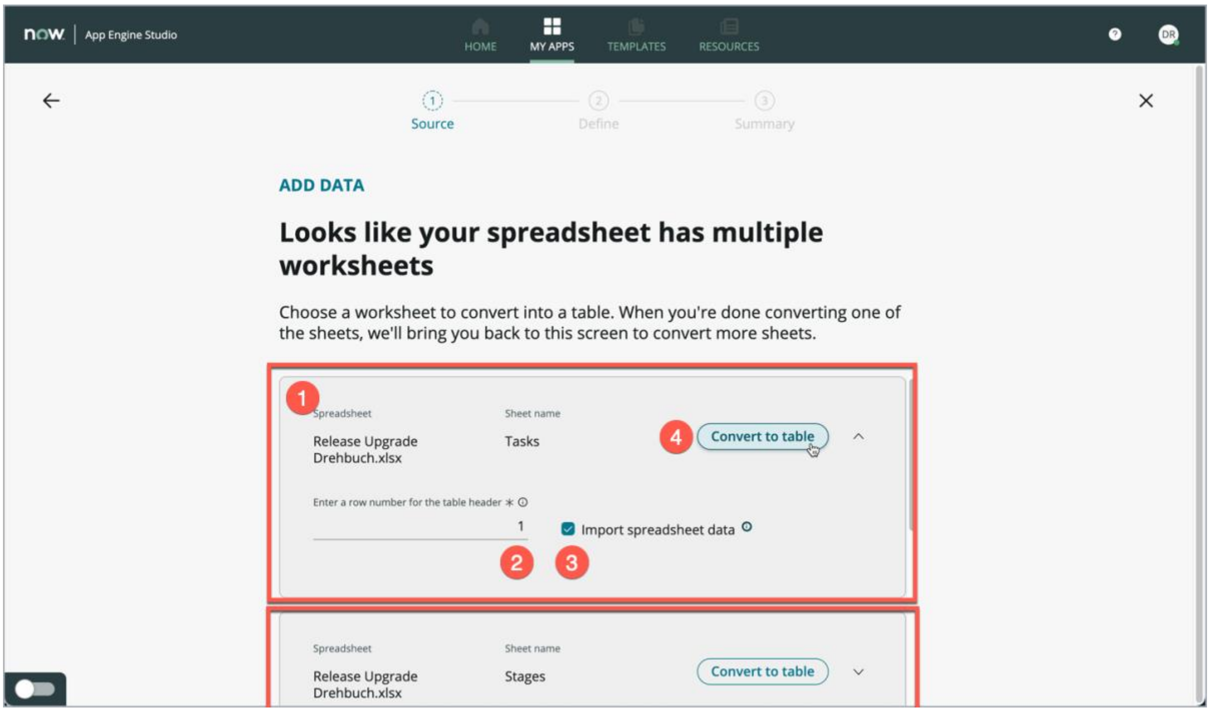
The filename will be mirrored in the dialog (1). Click "**Continue**" (2) to move on.

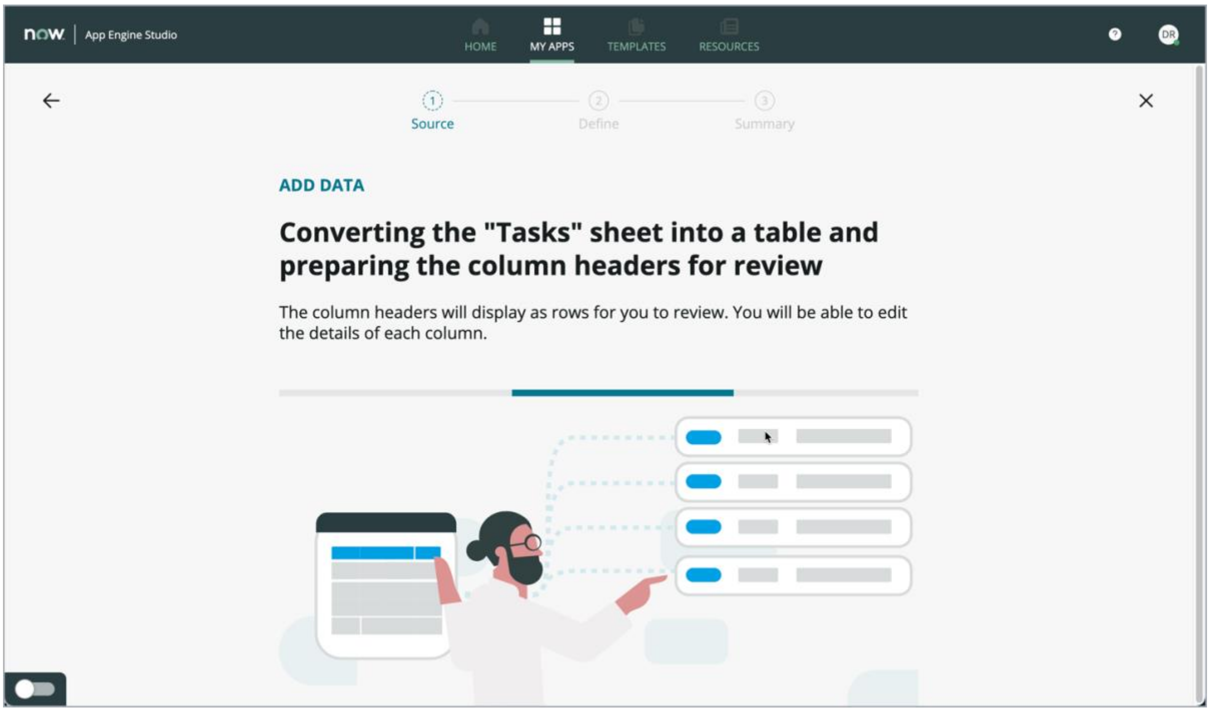


Note:
As soon as the Excel file you import, has more than one tab, there will be the same number of boxes shown in the next screen. For our use case, we only want to import the first tab (1).

Mark the "**Import spreadsheet data**" (3) Checkbox and review, that the field names will be derived from the first row (2) of the Excel file.

Click "**Convert to table**" (4) to move on.

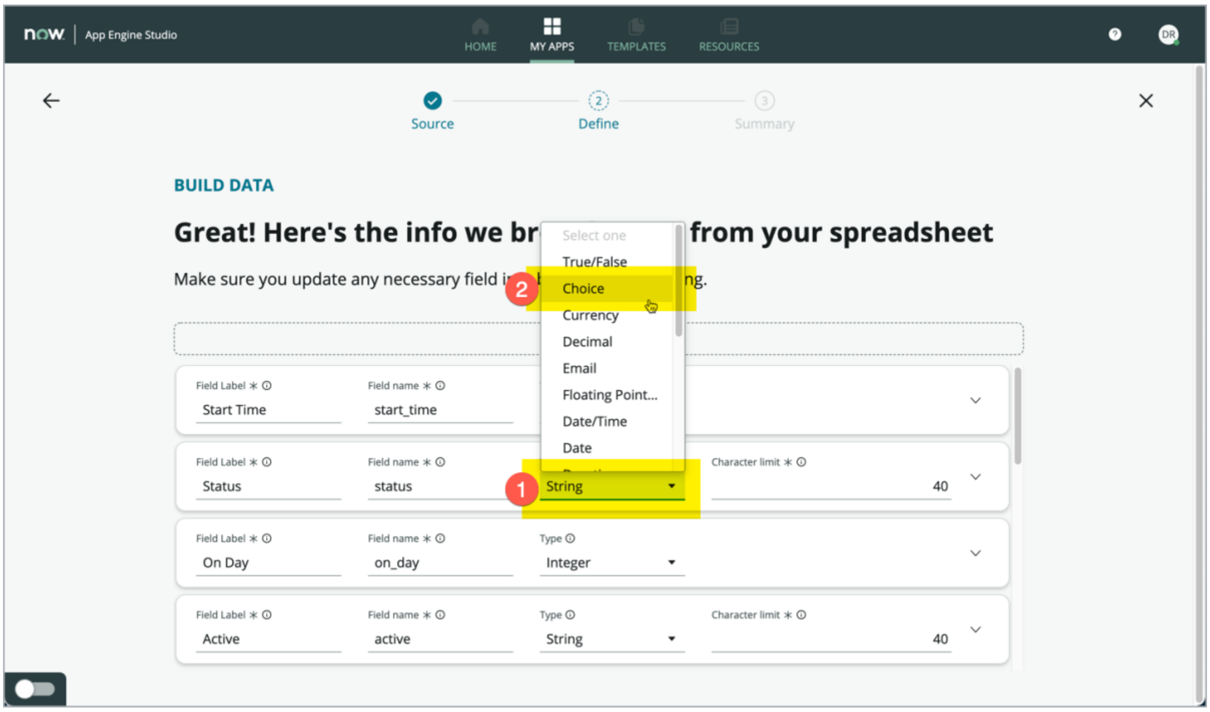




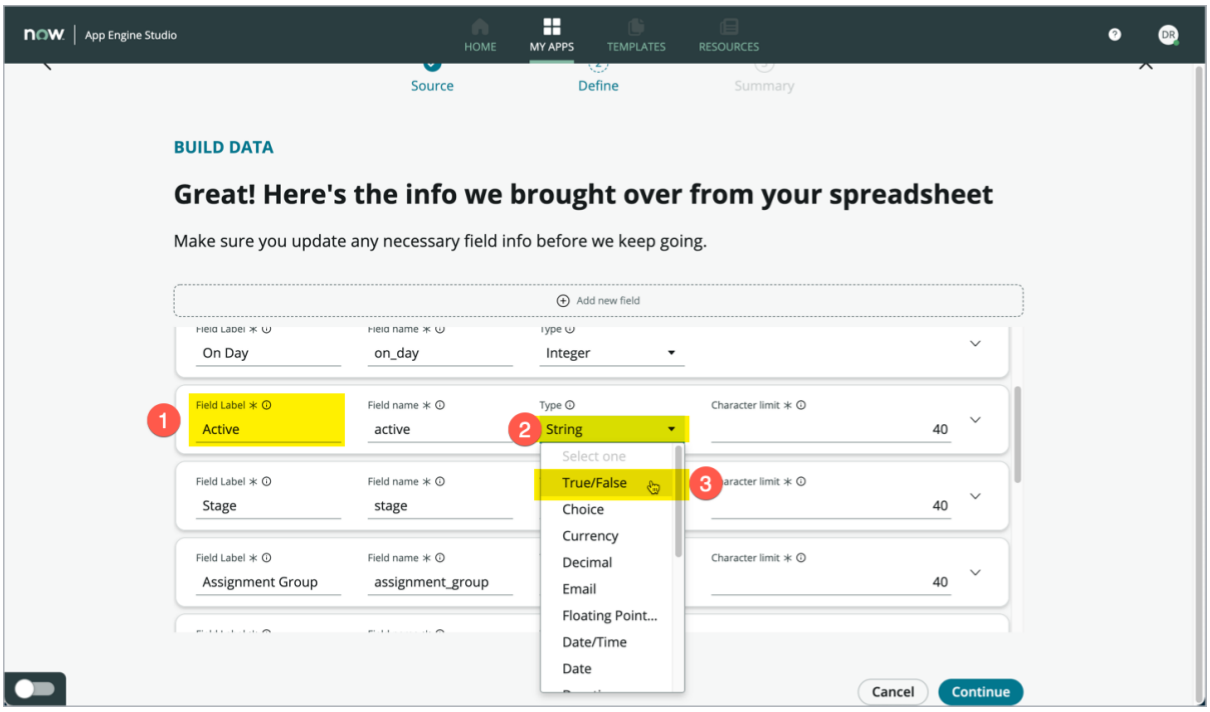
In the next step, you can review the Columns and fields identified in the Excel spreadsheet.

Now it is time to easily adjust field types and sizes, as well as field names – if desired.

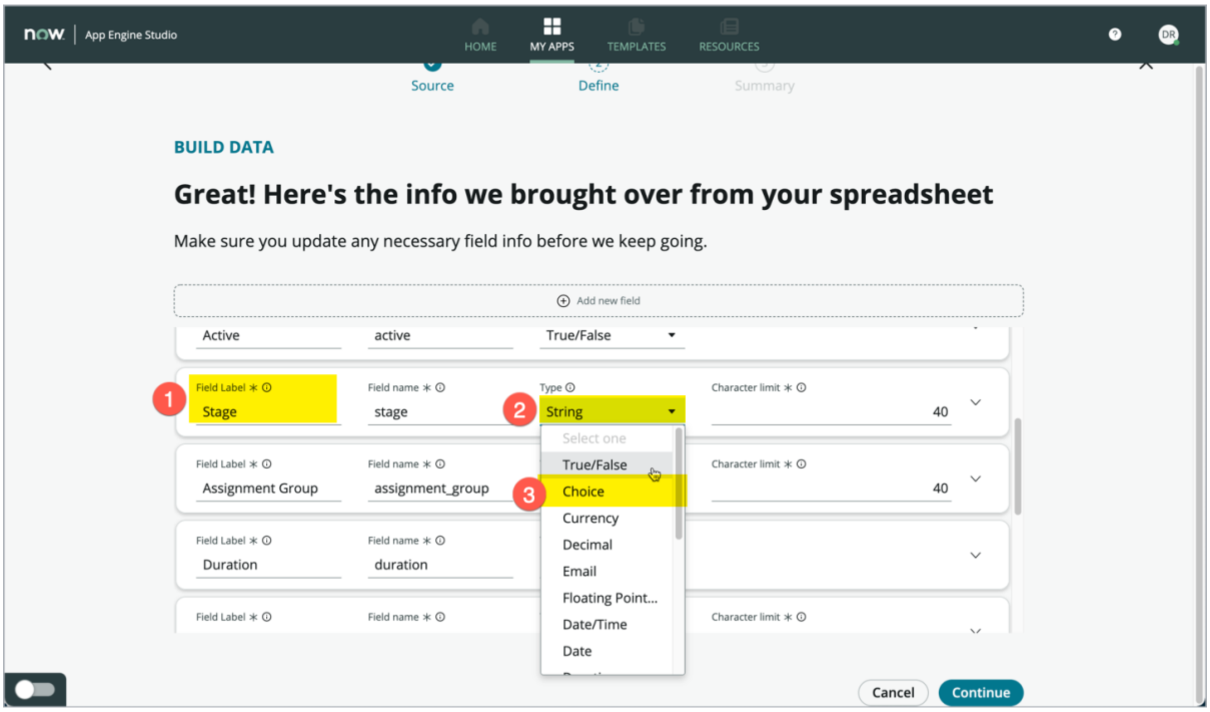
Click the "**Type**" dropdown of the **"Status" field** (1) – which was defaulted to a "String" type field – and **Select "Choice"** (2) from the List. This will define the new column of the field as dropdown as well.



Scroll down and find the **"Active" field** (1). Click the "**Type**" dropdown (2) and select "**True/False**" (3). This will result in a **Checkbox style** user interface field.

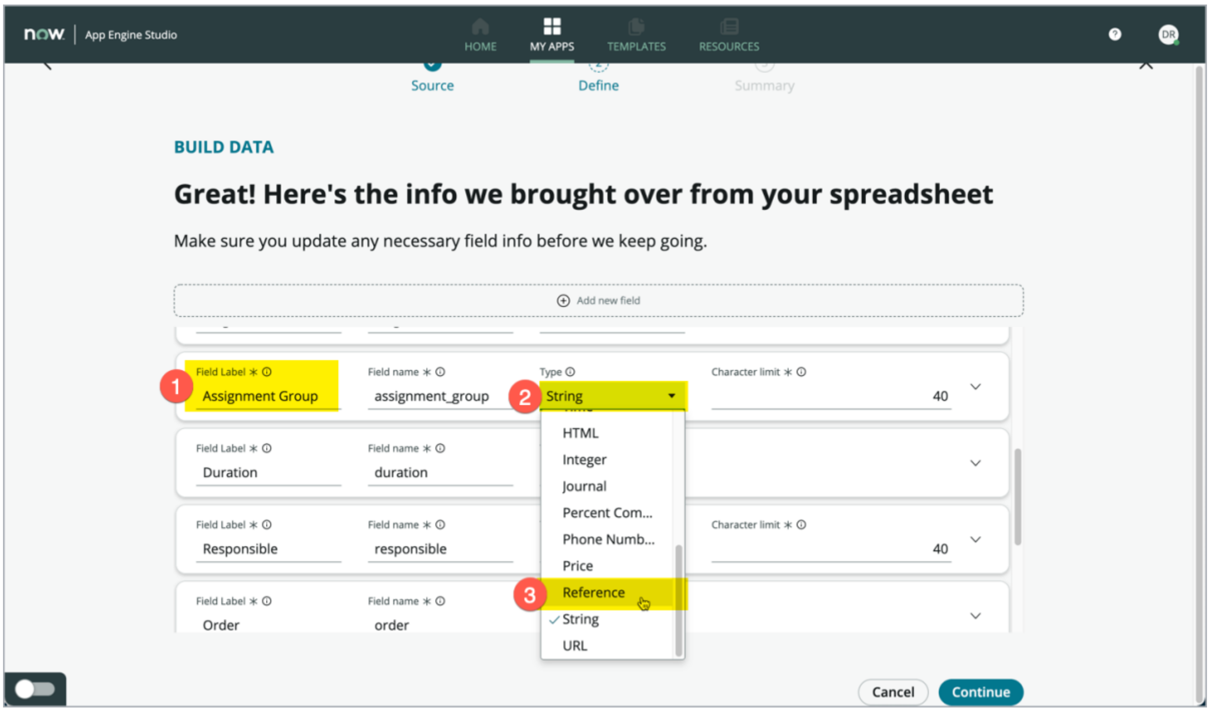


Same for the **"Stage" field** (1). Change the Type from "String" (2) to "**Choice**" (3):



Note:
Selecting the column type "Choice" will automatically populate the Choice options for you, by using all unique combinations/values found in the source spreadsheet file for that given column.

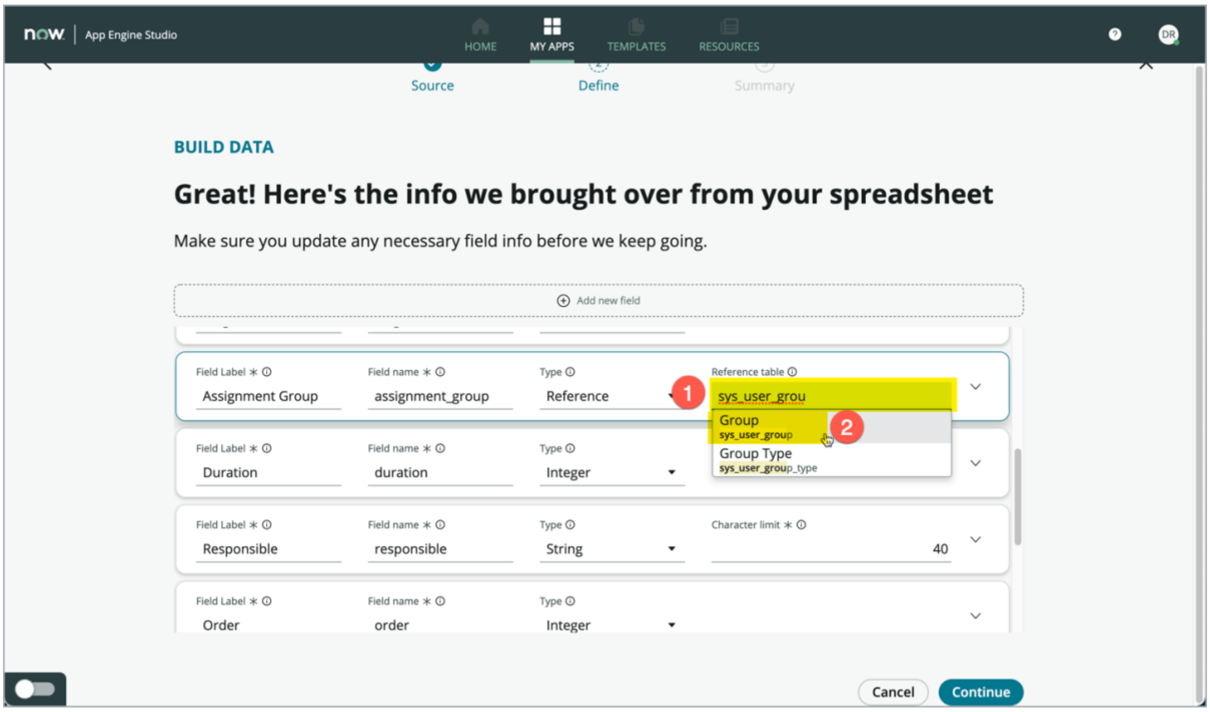
Change the **"Assignment Group" field** (1) from "String" type (2) to "**Reference**" type (3).



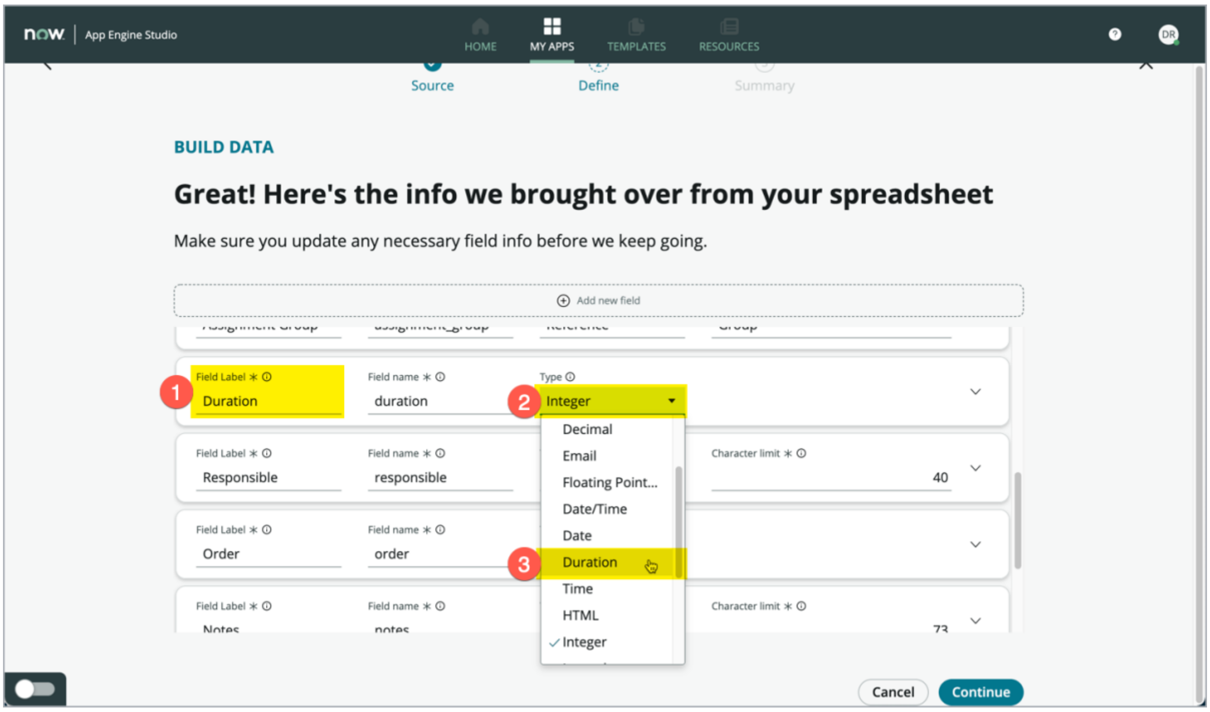
For Database fields of the **Type "Reference"**, you need to define, to WHICH table the reference is defined. Therefore, you need to fill in the "**Reference table**" (1) field.
From the List of tables, select the "**Group**" **[sys_user_group]** table.

Note:
The fastest way to find this table is by filtering the list of tables. Just type in (part) of the searched table name (1), and the list of tables in the dropdown list will be filtered automatically (2).

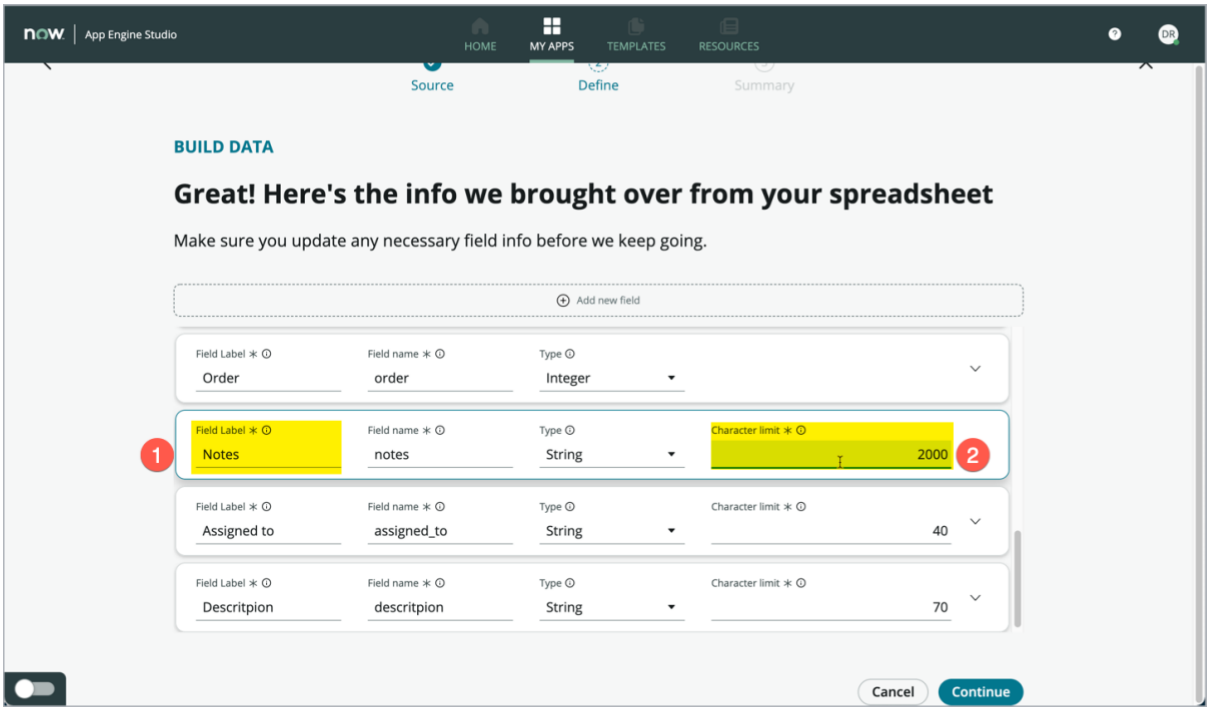
Make sure to exactly select the table name as shown in the screenshot below.



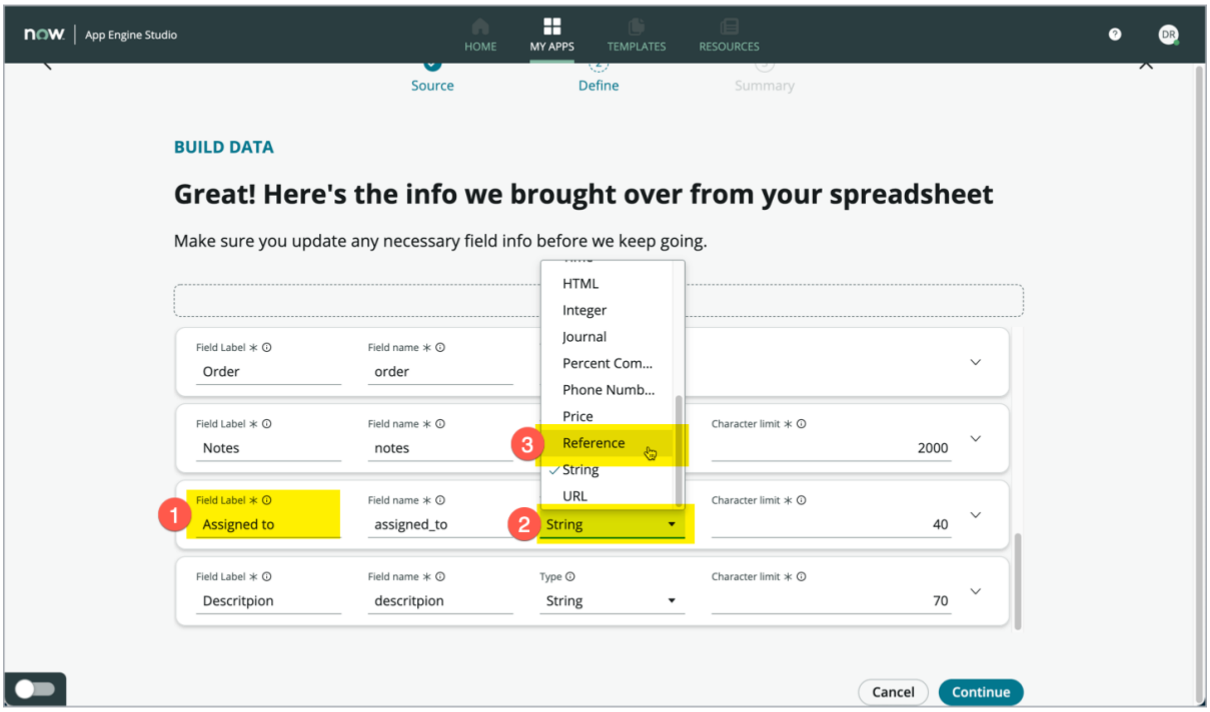
Change the field type of the **“Duration” field** (1) from “Integer” (2) to “**Duration**” (3).



Change the size of the **“Notes” field** (1) to **“2000” characters** in the right column (2).

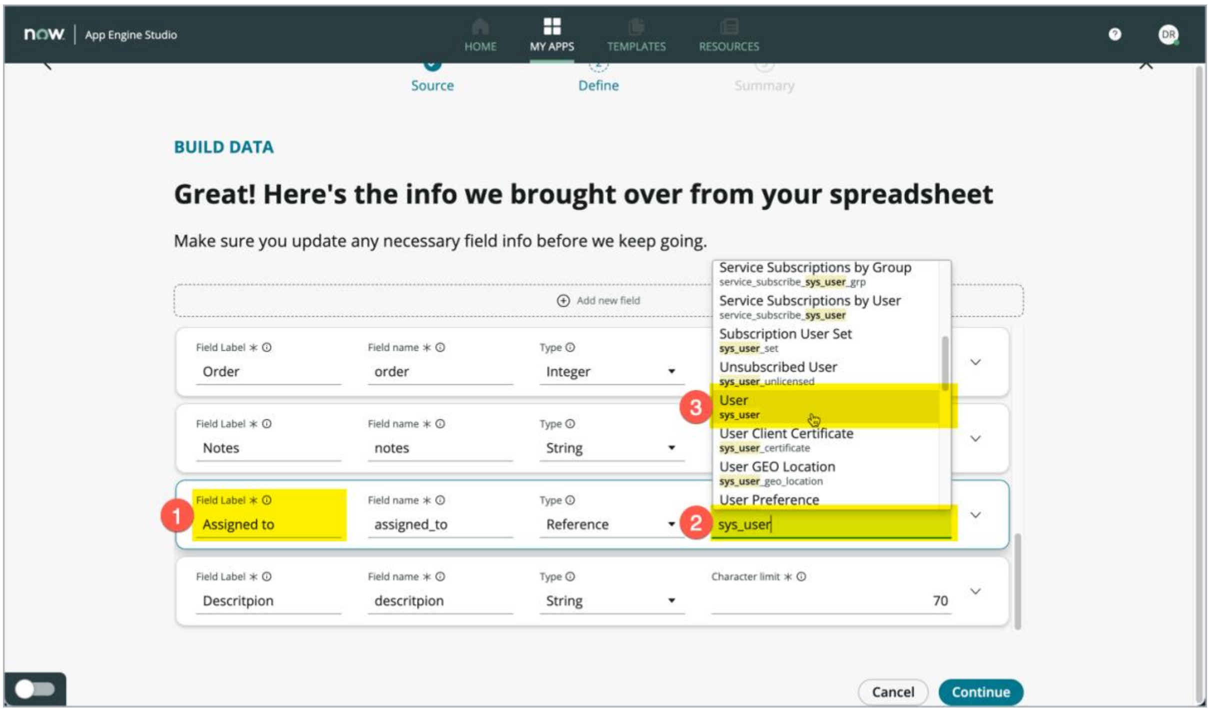


Change the field type of the **"Assigned to" field** (1) from "String" (2) to "**Reference**" (3)

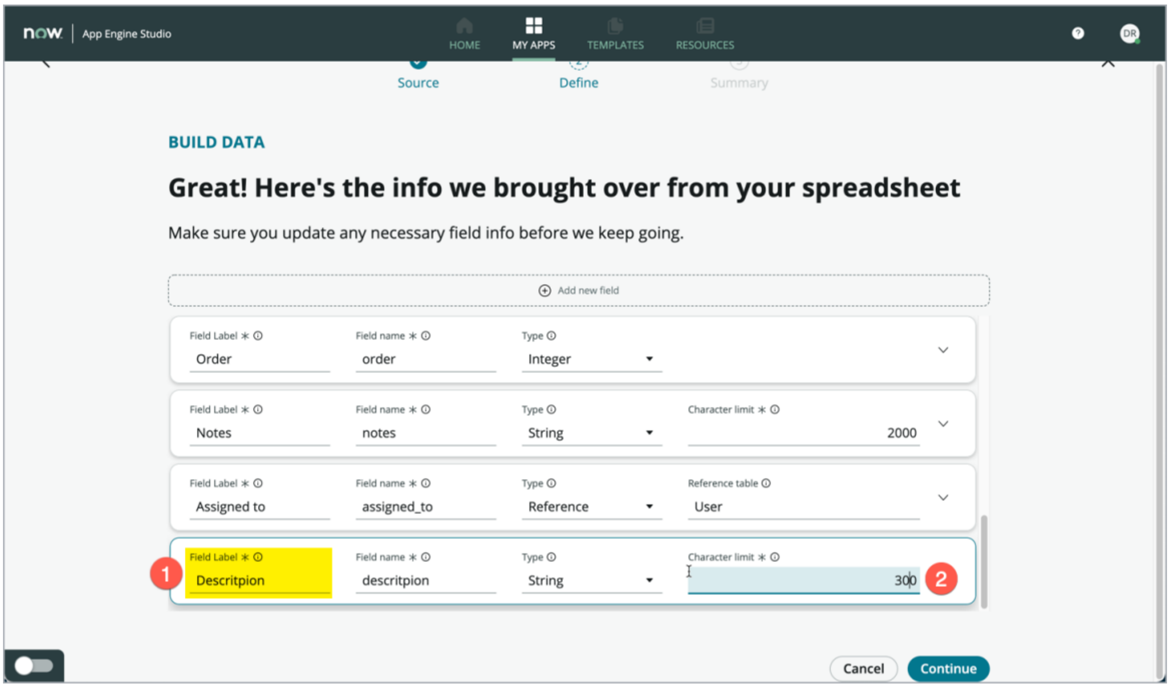


And finally, set the **Referenced table** for the **"Assigned to" field** to the **"User" [sys_user]** table.

Note:
Find and use the table exactly as shown in the screenshot below, as there is a high number of tables containing "user". Filter the list before scrolling it.



Expand the field size of the **“Description” field** (1) to “**300**” characters (2).

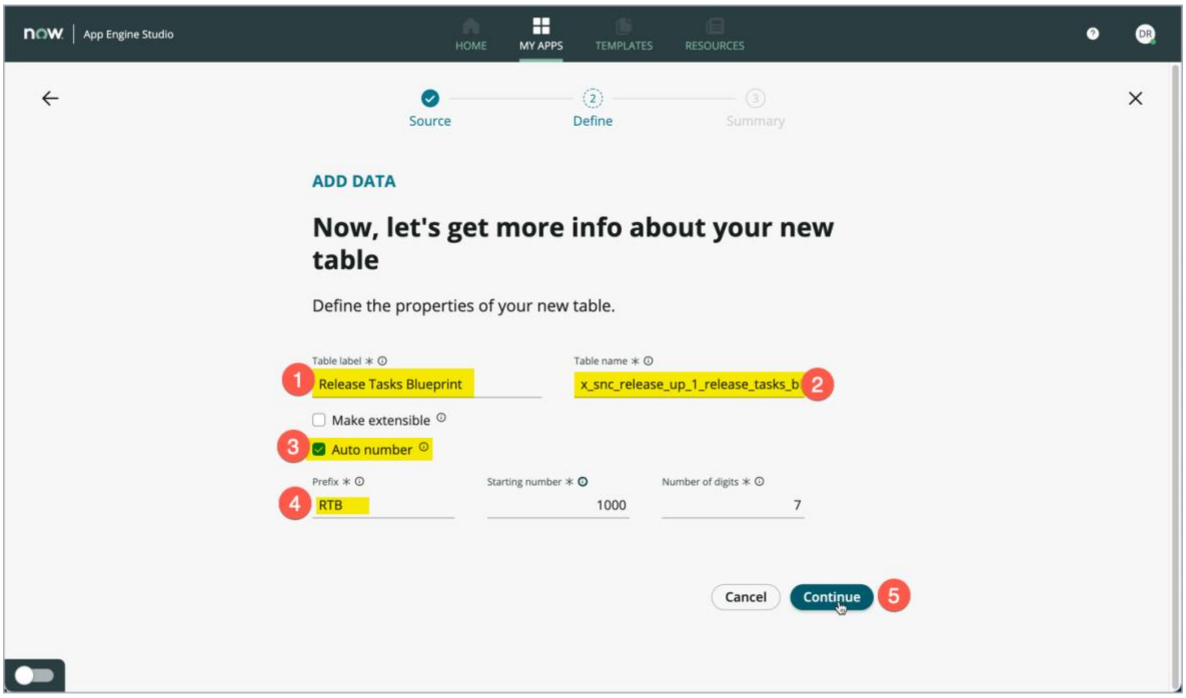


Hit “**Continue**” to move on

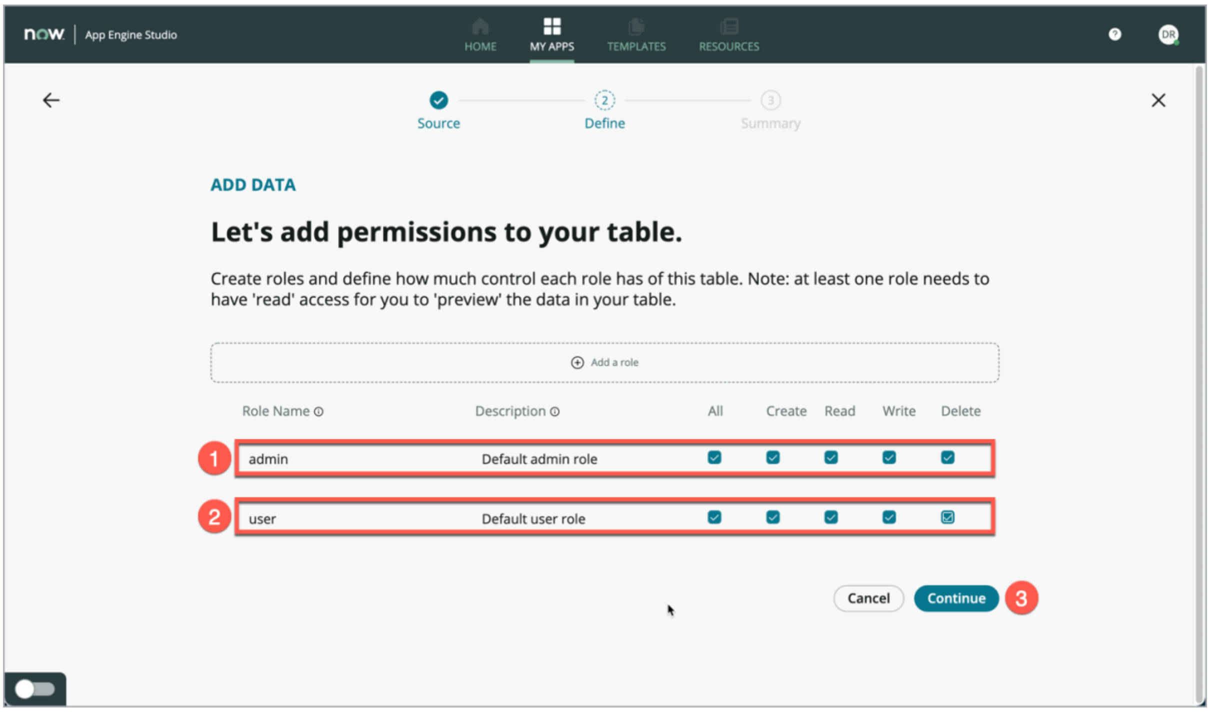
Next, we name the table (like we did for the first two tables we created).
Set the "**Table label**" (1) to "**Release Tasks Blueprint**" and don't change the "Table name" (2) field.

Mark the "**Auto number**" Checkbox (3) and set the "**Prefix**" to "**RTB**" (4).

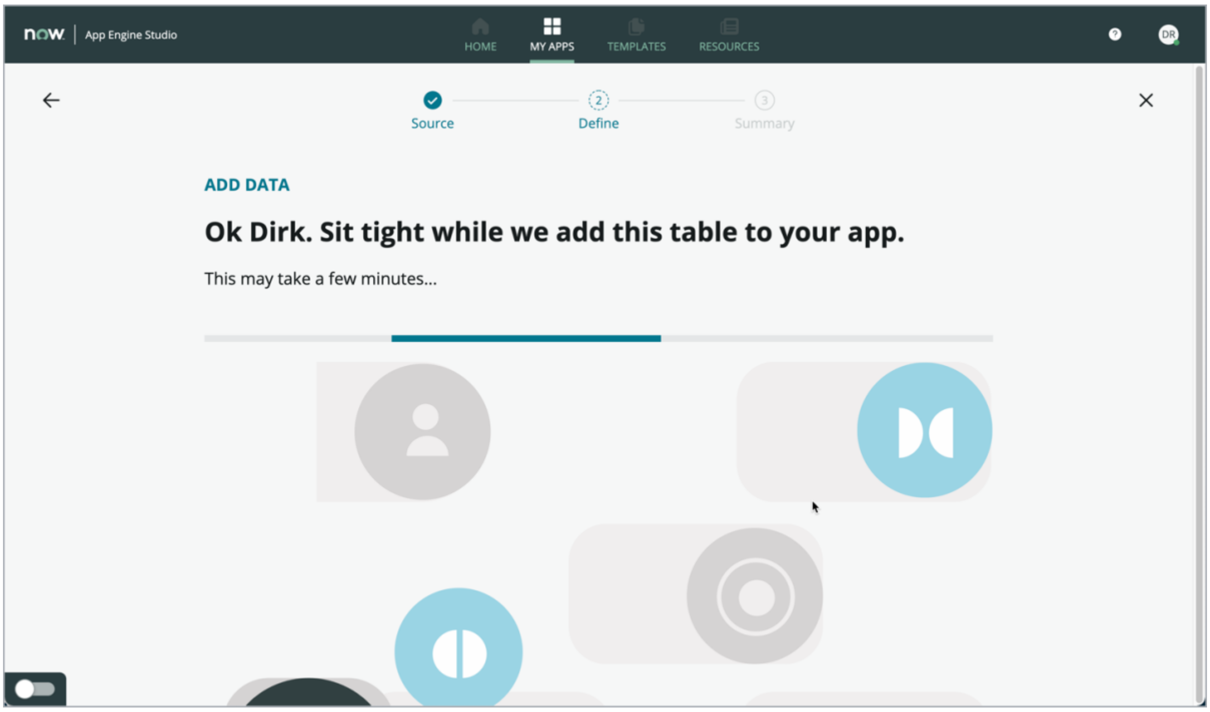
Click "**Continue**" (5) to move on.



Give **full access to all Roles** (1) & (2) and click "**Continue**" (3)



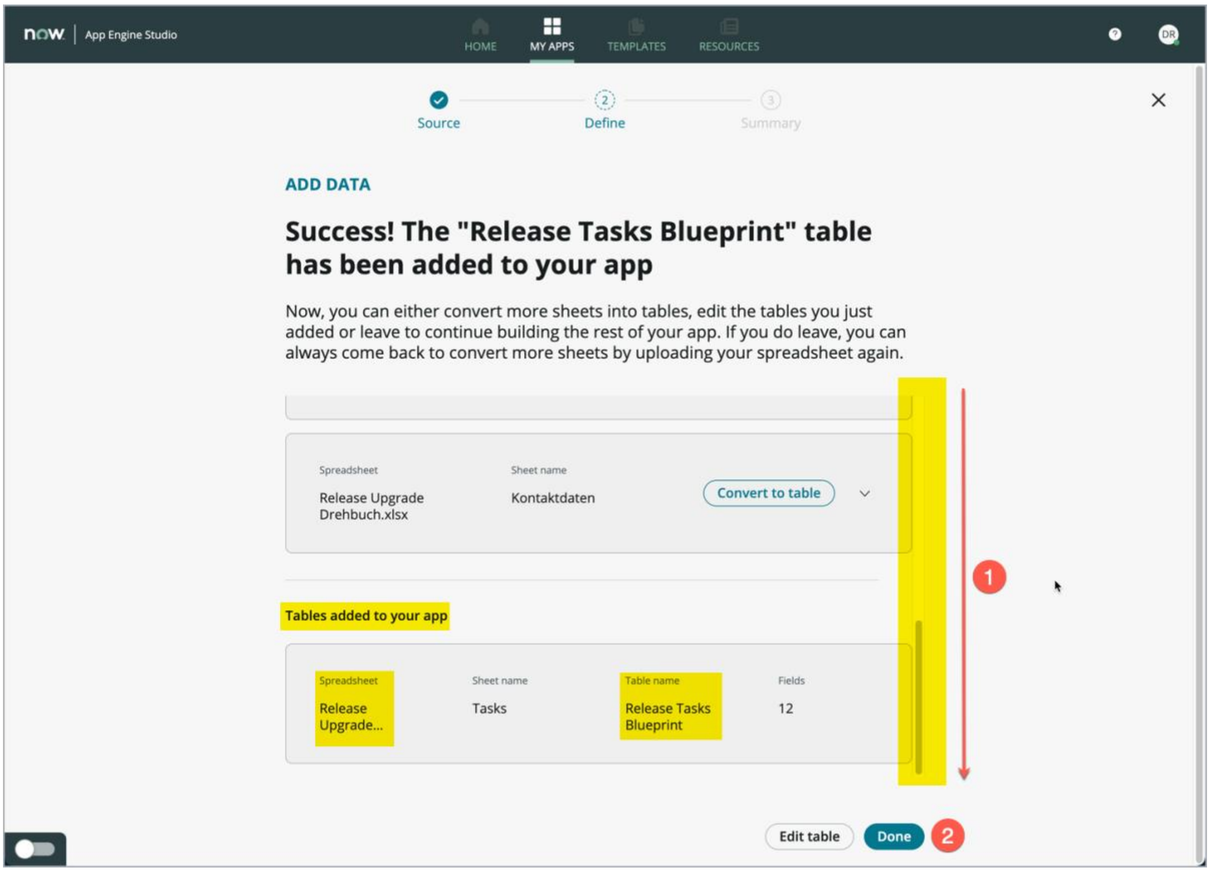
ServiceNow reads the Excel file, creates the table and fields for you and in the same moment, also imports the data from the Excel file into your ServiceNow instance.



The wizard now allows you to import even more tables from other tabs of your spreadsheet, but we do not want to import more data.

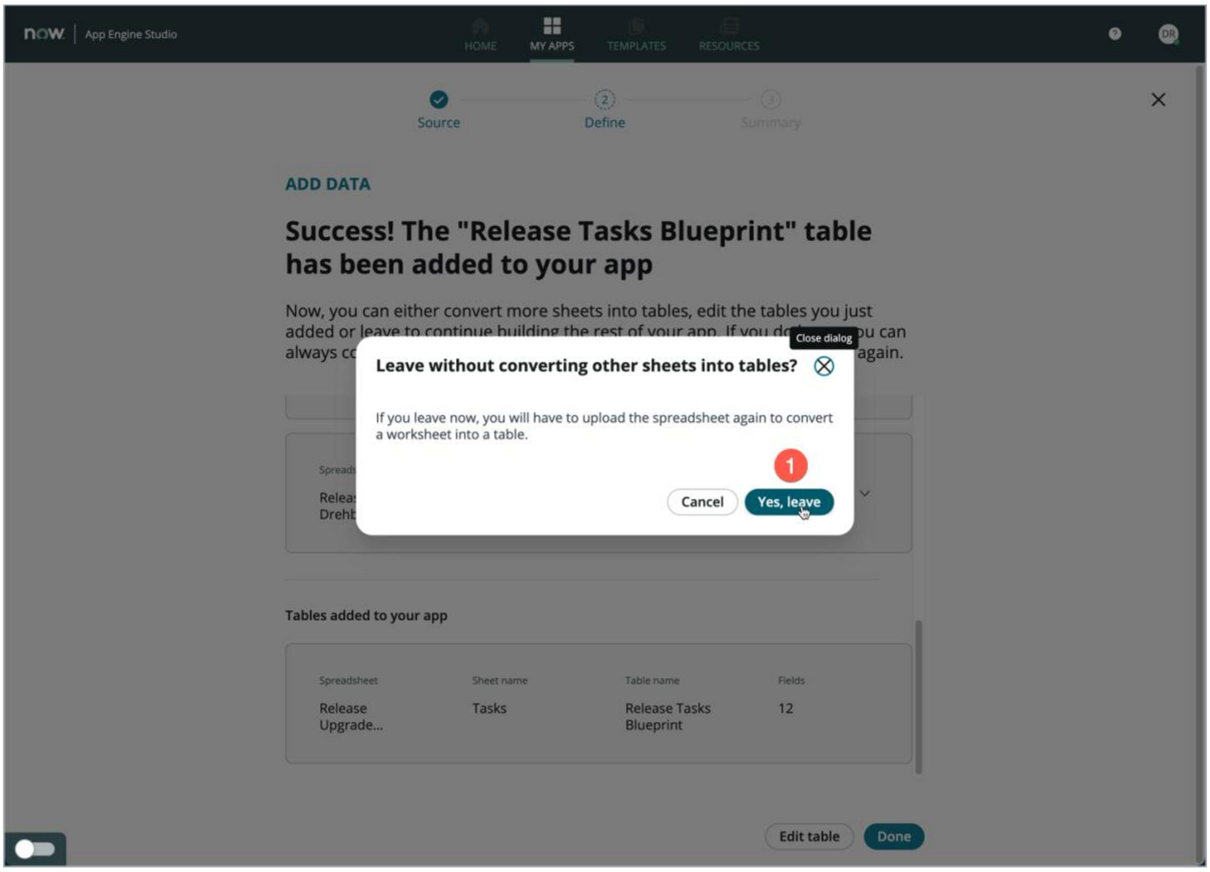
Scroll down the screen (1) and click "**Done**" (2)

Note: You can see the already imported tabs from your Excel file listed at the end of the list (see screenshot).

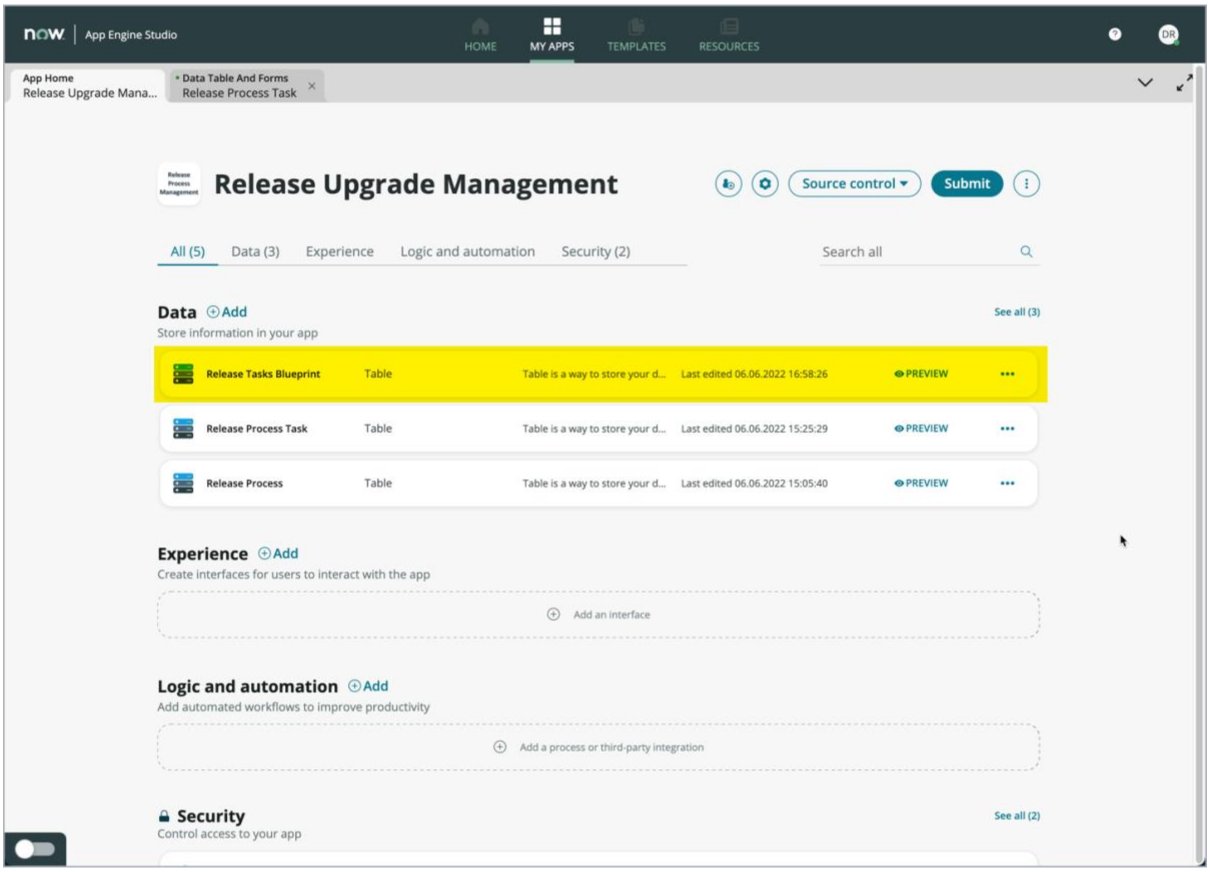


A message appears asking, if you want to leave as long as there are other tabs of the Excel file, which have not been imported yet.

Click “**Yes leave**” (1) to move on.



Back on your App Dashboard, you can find the new table imported



Congratulations, you now have created the third table by uploading a spreadsheet.

This way to create tables is very fast, easy and efficient, while importing data in the same moment.

3 Configure Lists & Forms

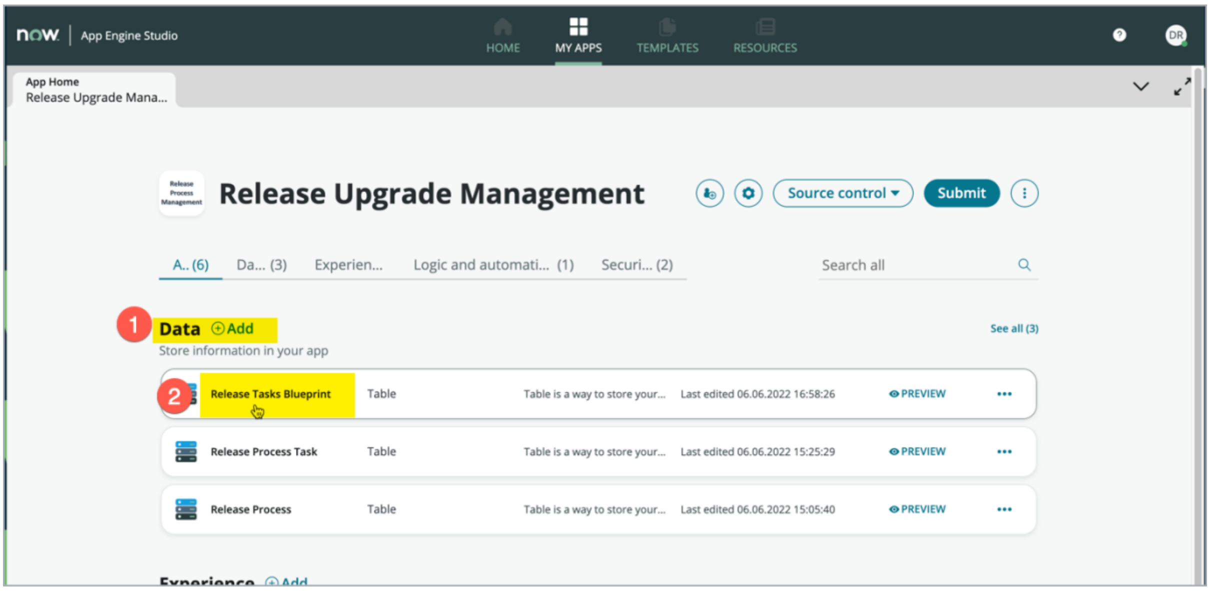
In this part of the Lab, we will configure the List Views and Form Views of the three tables created in the previous Exercises.

3.1 Exercise #3 – Configure the Layout for the List & Forms

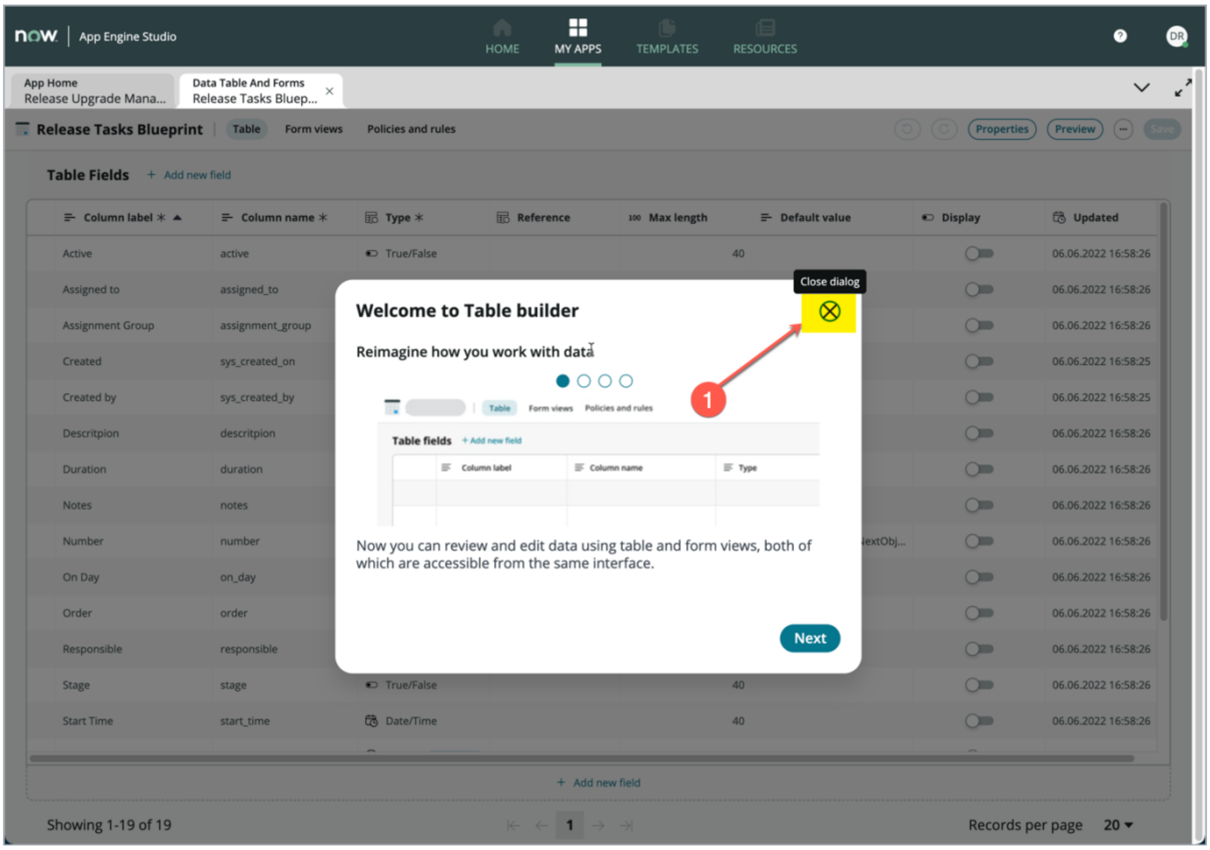
We start from the Dashboard of your App in App Engine Studio.
There are different options in the platform to configure Lists and Forms. For today, we will start from Table Builder within App Engine Studio.

3.1.1 Layout for the "Release Tasks Blueprint" table

In the "**Data**" section (1), click the name on the left side for the "**Release Task Blueprint**" table (2).



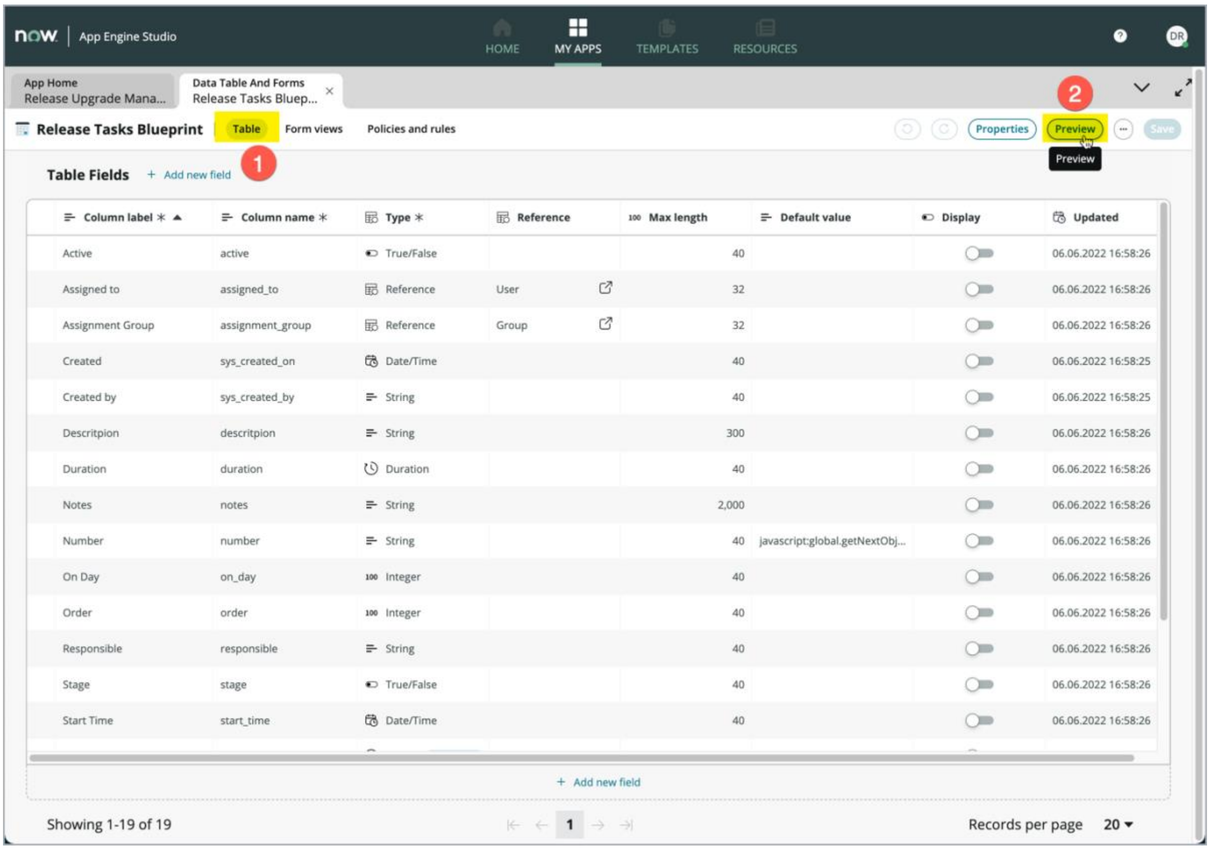
Close the Popup dialog by using the X-sign in the upper-right corner (1).



a) Configure the LIST layout

The first page you see from Table Builder is the "Table" Tab (1). This is the tab, where you can manage all columns/fields from the given table.

Click "Preview" (2) to open a new Tab in App Engine Studio to preview the List View for the "Release Tasks Blueprint" table.

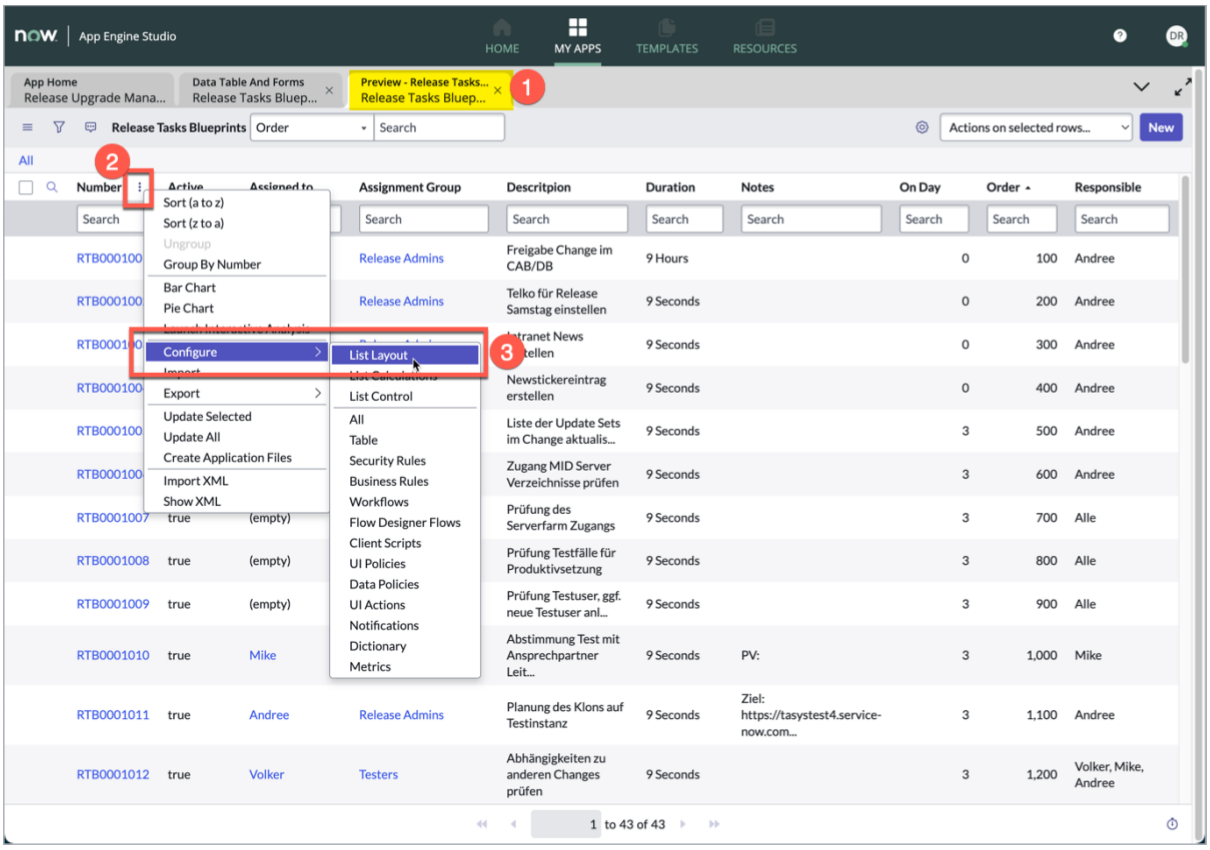


As you can see in the screenshot below, there is a new Tab in App Engine Studio (1), for the List View of the “**Release Tasks Blueprint**” table.

Hover the mouse over the column header of the “Number” column, so that the small menu icon with the three vertical dots appears (2).

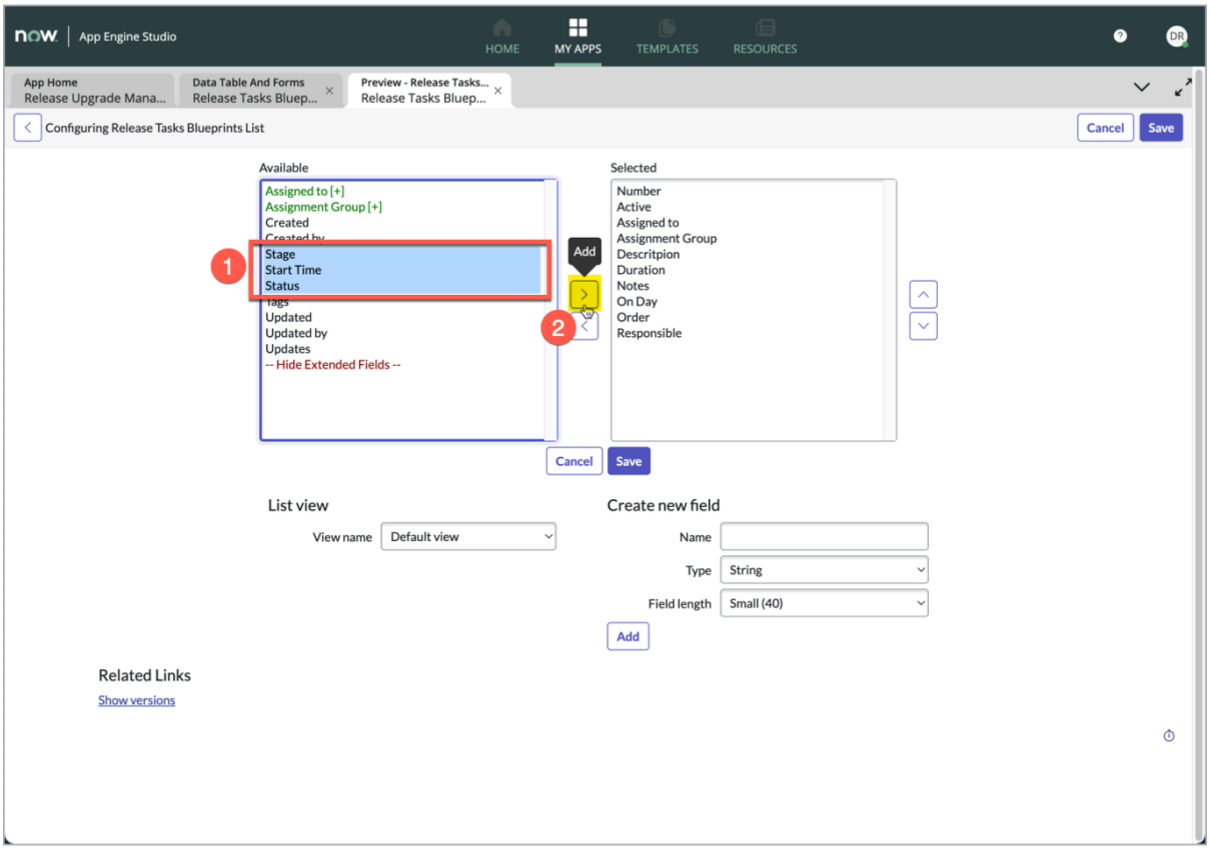
Click that icon and open the menu of the List.

Select the “**Configure**” Entry and then “**List Layout**” (3).

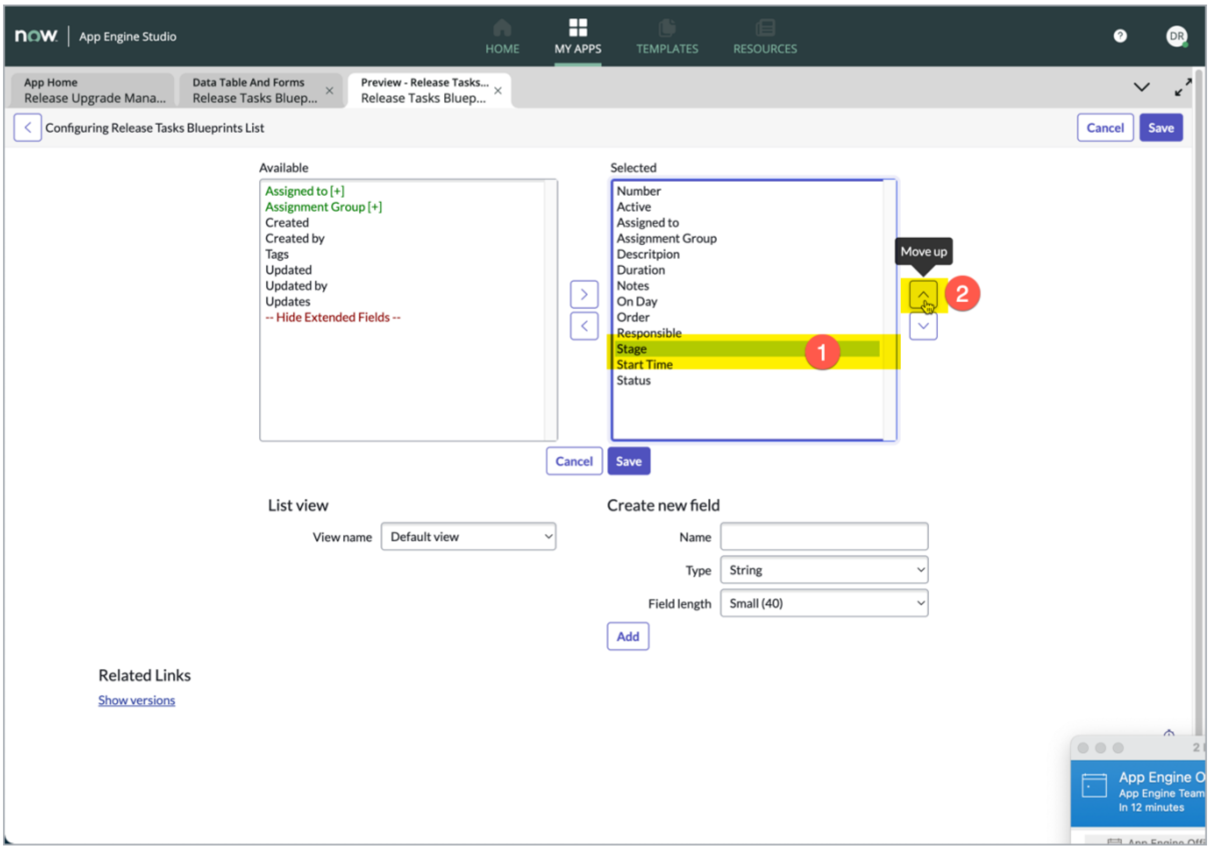


The left list is showing fields which are currently NOT in the List view but are available to be added to the List. Select (also multiselect is possible) items in the left list (1) and click the "Add" (">") button (2) to move the fields to the list on the right side. The right-hand list shows the columns top-down that appear in List View, in the order from left to right.

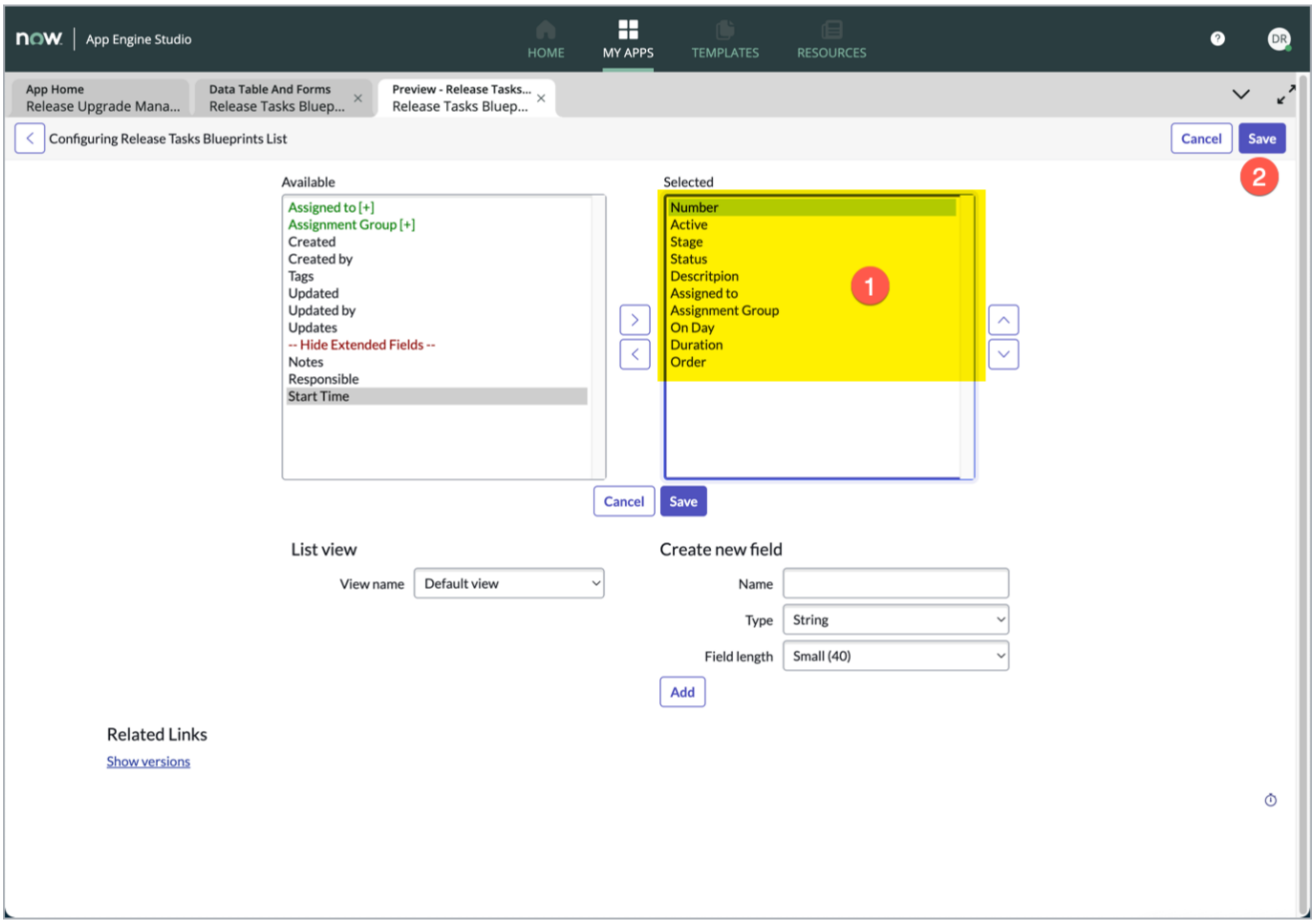
You can also remove fields from the "Selected" List (on the right-side), by selecting the entries in the list and click on the "Remove" ("<") button (2).



Select one or more columns in the right-hand list and use the "Move up" and "Move down" buttons (2) to change to order of the columns in the List View.

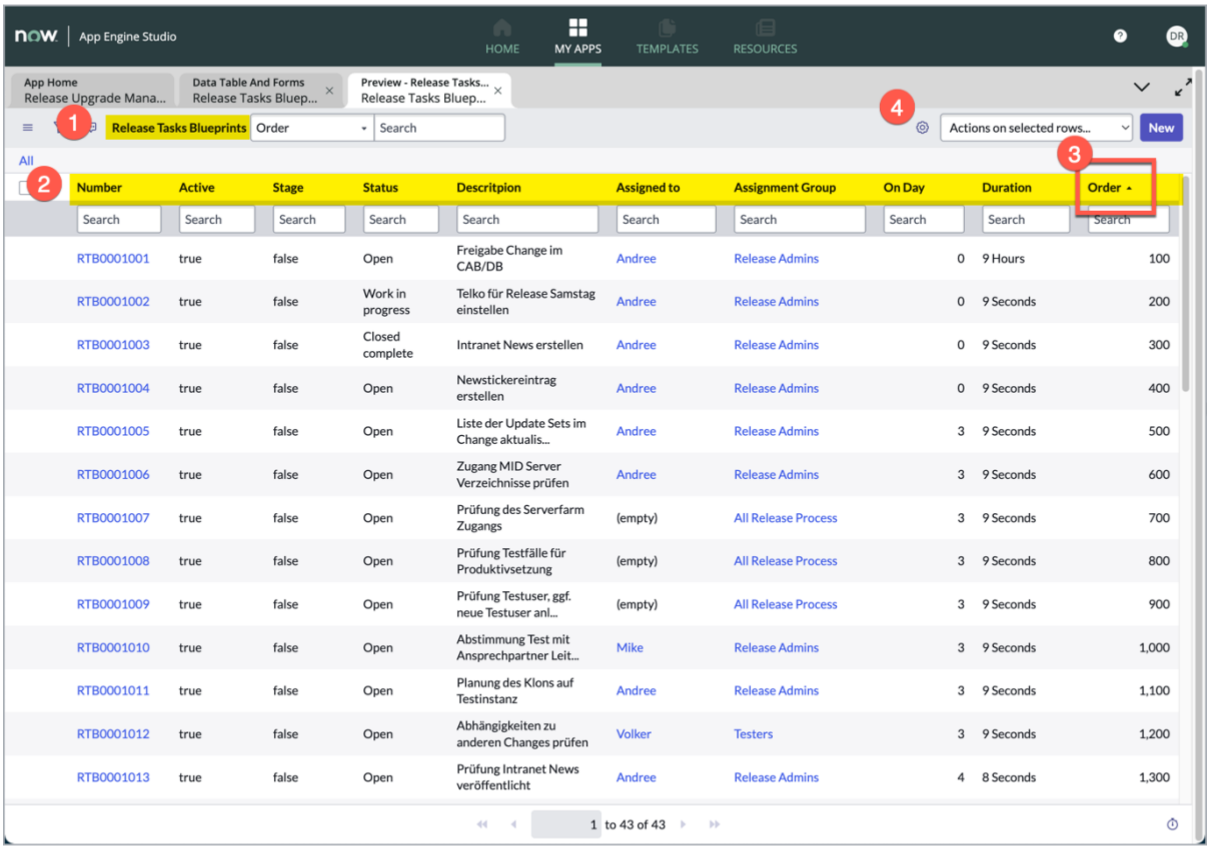


Try to arrange the fields as shown in the screenshot below (1) and click "Save" (2) when done.

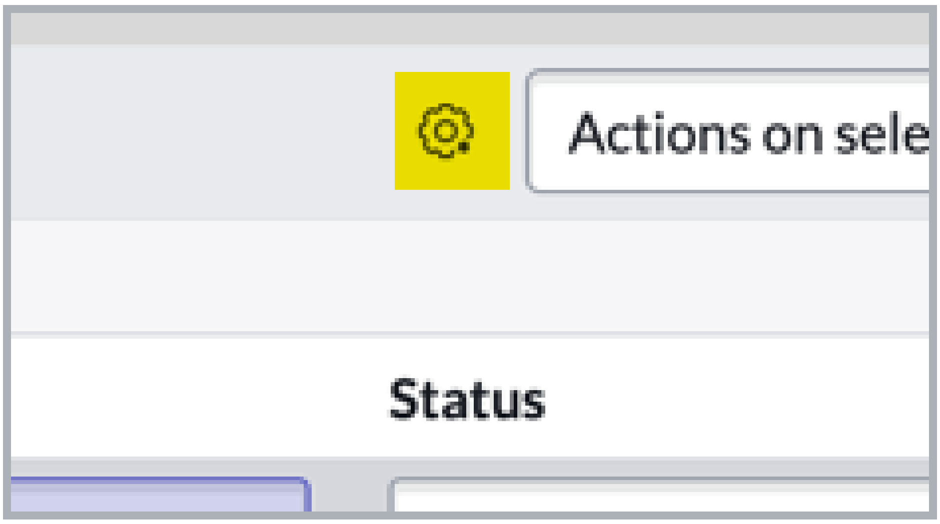


You will navigate back to the List view (1) and the columns should now appear in the order you defined (2). Note that you can also sort the list ascending or descending, just by clicking on one of the column headers.

For users, there is additionally the option to "Personalize" the list (4). Clicking the small gear icon opens a similar dialog like shown above, but this time, you configure the layout just for yourself. That overrides the global settings configured above.



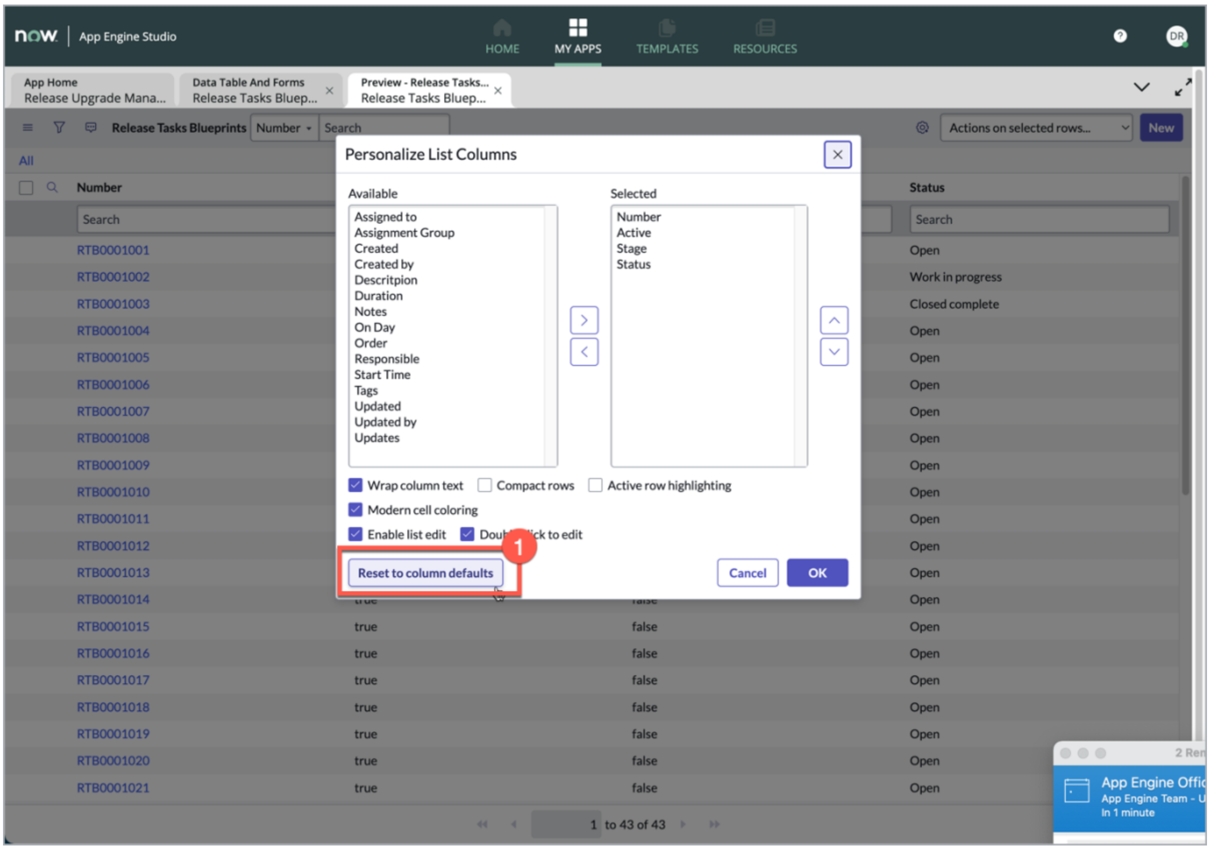
As soon as you have "personalized" the List Layout, there is a small dot at the gear icon (see screenshot below). That indicates that the List view is personalized and may not reflect the standard layout we configured above (by the System Administrator).



Click the small gear icon again to re-open the "Personalize List" screen – see screenshot below.

As soon as you have a custom (personalized) Layout of the Form, the button "**Reset to column defaults**" (1) is shown. Click that button to remove your personalized layout for this List View and revert to the Standard layout defined by your System Administrator.

Note:
Whenever the layout does not reflect the settings of the List layout you configure, review first if there are some personalized List layout that you can reset/revert.



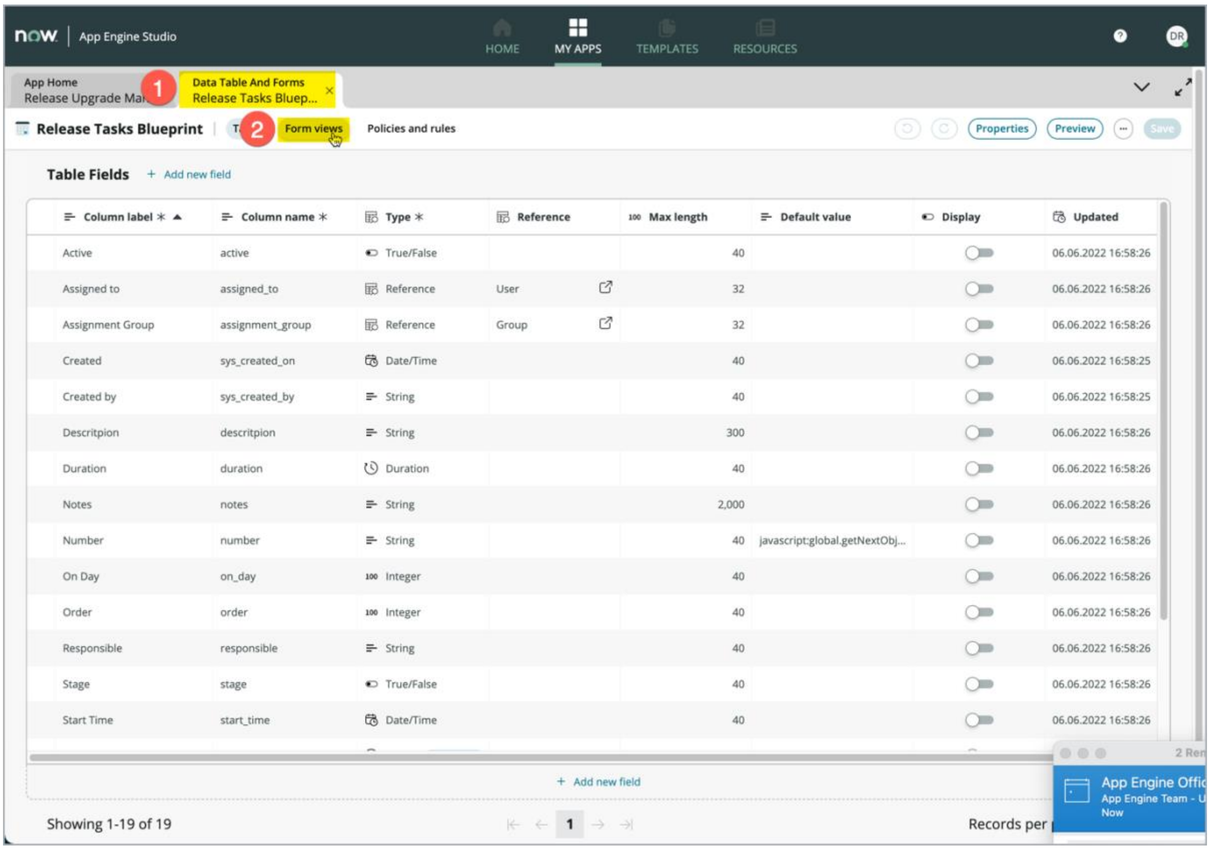
Congratulations, you just configured your first List Layout.

b) Configure the FORM layout

Next, let's configure the **Form layout** for the "**Release Tasks Blueprint**" table.

Close the List tab in App Engine Studio, so that you come **back to the Table Builder** opened on the "Release Tasks Blueprint" table (see screenshot below).

Make sure, that you are on the Table Builder Tab (1). Reopen it from the App Dashboard, if you closed it before. Having Table Builder open (for the "Release Tasks Blueprint" table), click "**Form views**" (2) to **switch to the next Tab of Table Builder**.



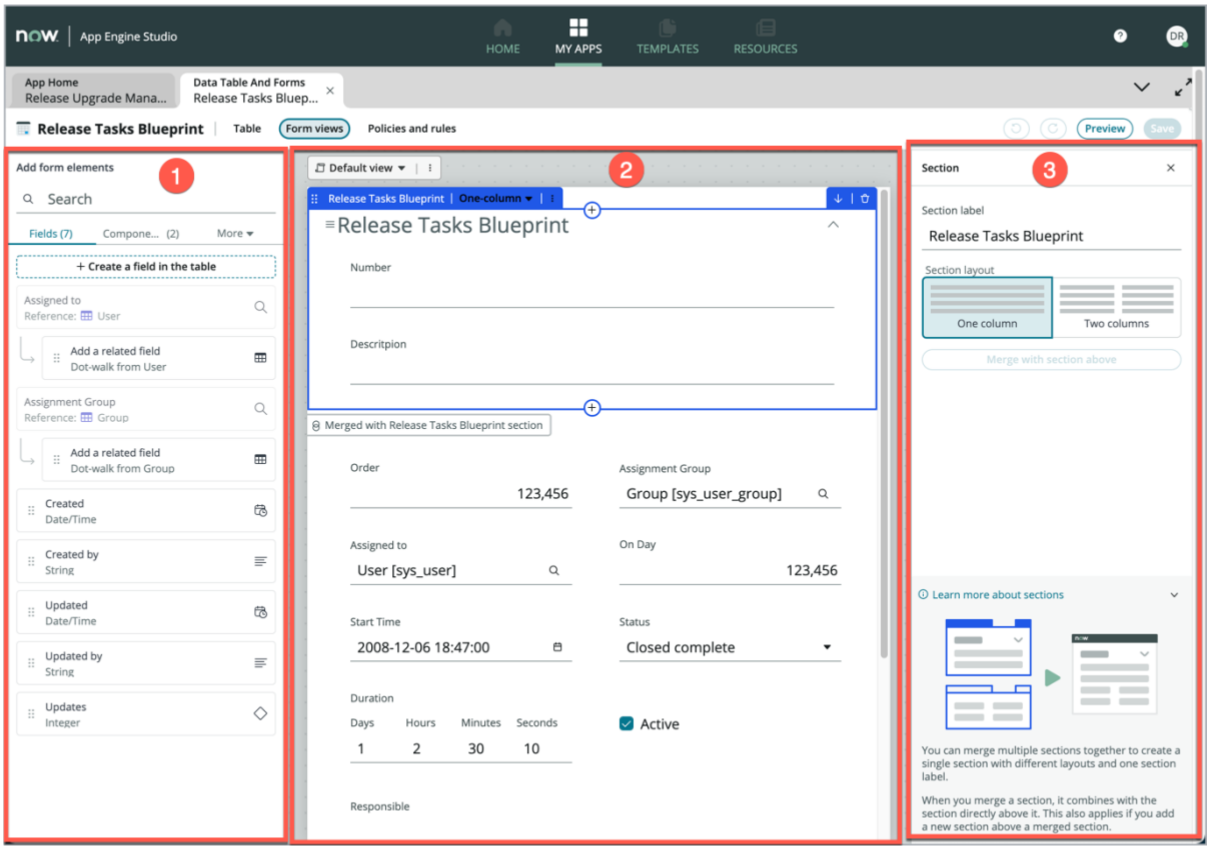
The screen has three main areas.

Left (1), there is a panel with form elements to be added to the form in the middle. The panel contains four tabs (“Fields”, “Components”, “Formatters” and “Embedded Lists” – found below “More”). Drag & Drop elements from here to the form space in the middle.

In the **middle** area (2), you can see the form in its current layout. You can arrange fields on the form by dragging them to the place where you want them to appear. Forms are constructed by sections, and each section can be either “One column” or “Two column” layout.

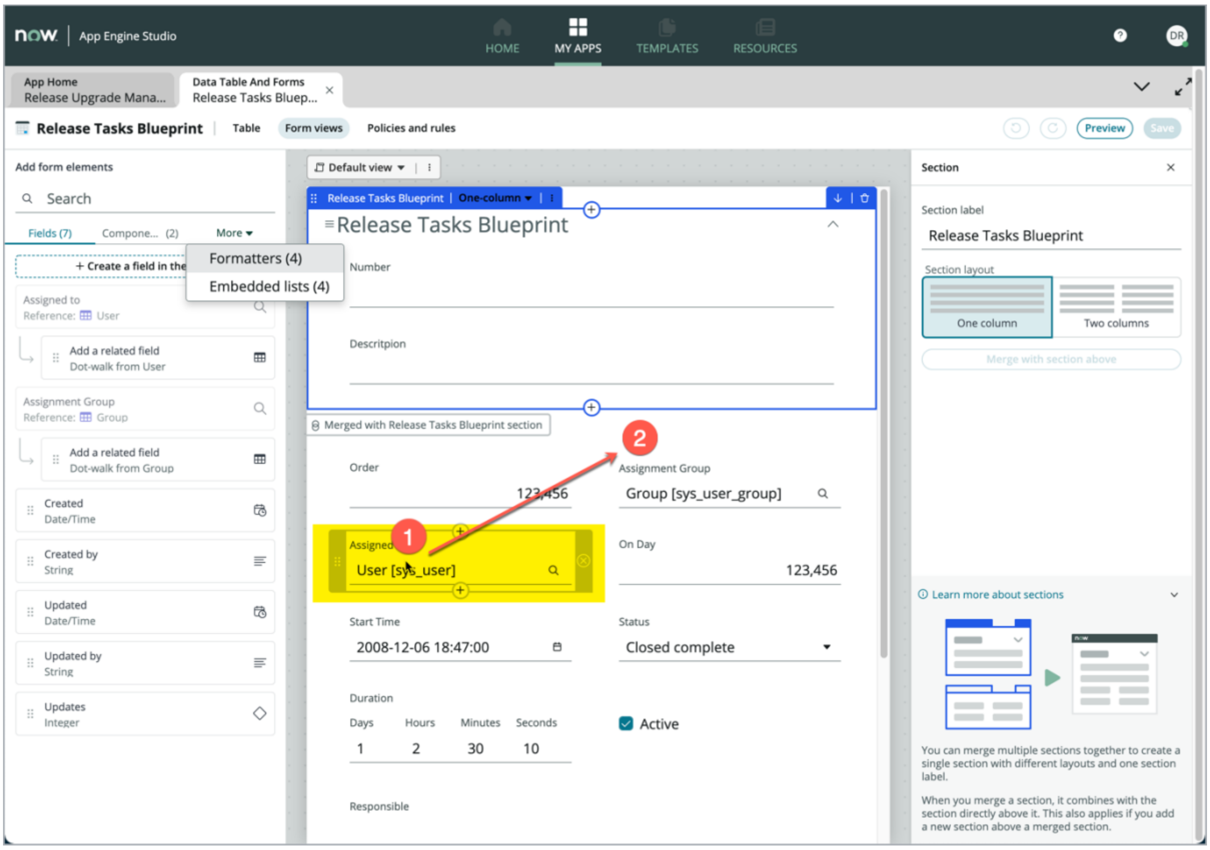
On the **Right** side (3), there is a panel to configure elements selected in the middle area.

There are a lot of configurations you can do here in this Builder. For detailed information, please consult the ServiceNow online documentation.



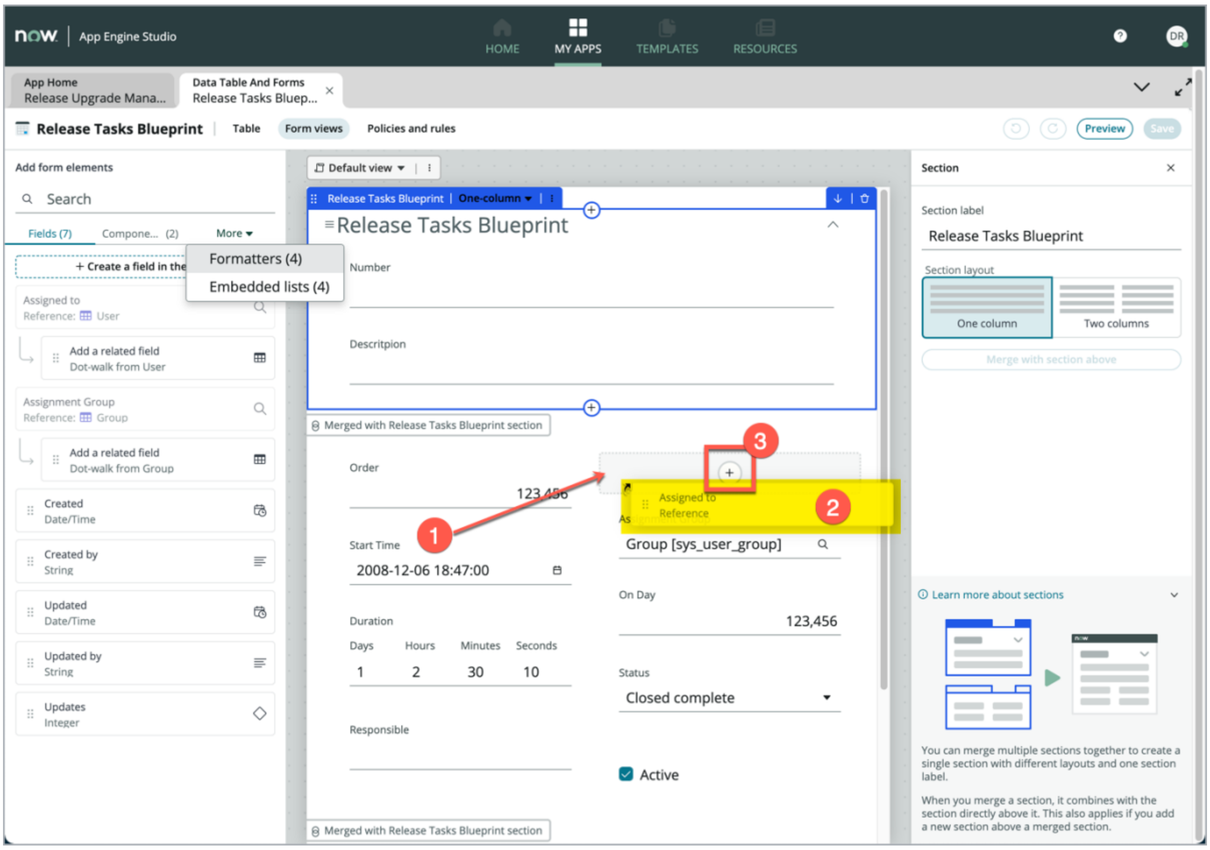
Let’s arrange the fields on the form in the way we would like them to appear.

Left click a field on the canvas (1) and **hold mouse button pressed**, so that you can **drag and drop** the column to a different place on the form (2).

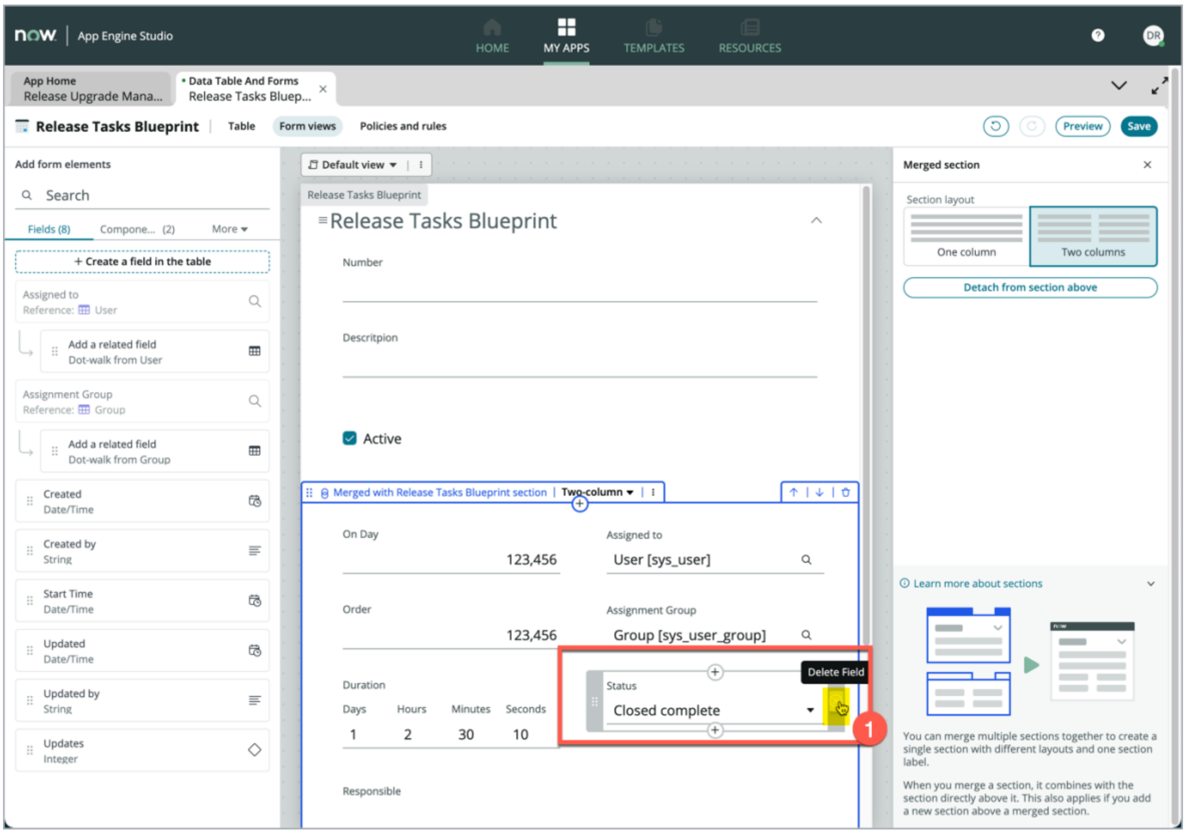


Moving the field on the canvas (1) will **show free fields** (3) where you can drop the field (indicated with the (+) Plus sign), that you are currently dragging (2) over the form.

Just release the left mouse button over a free space (having the plus symbol), to place the field on the form.

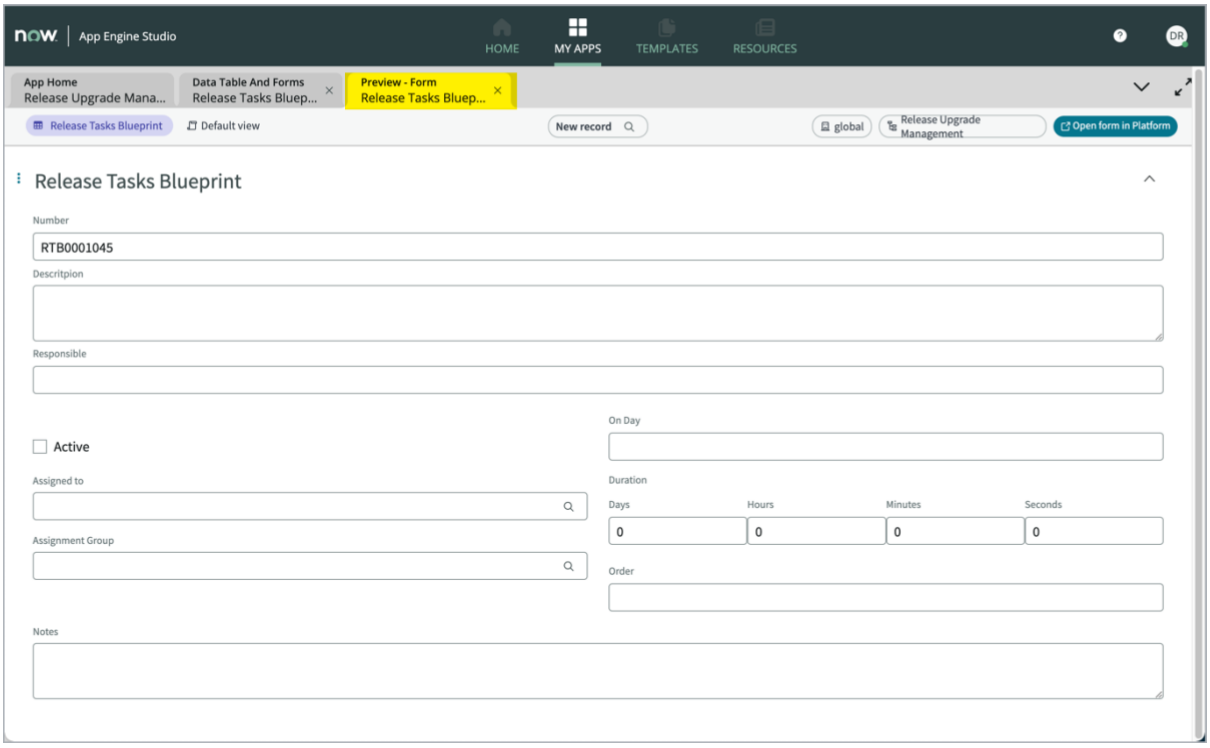


You can easily remove fields from the form by clicking the **small "x" symbol** (1) on the right side of each field.

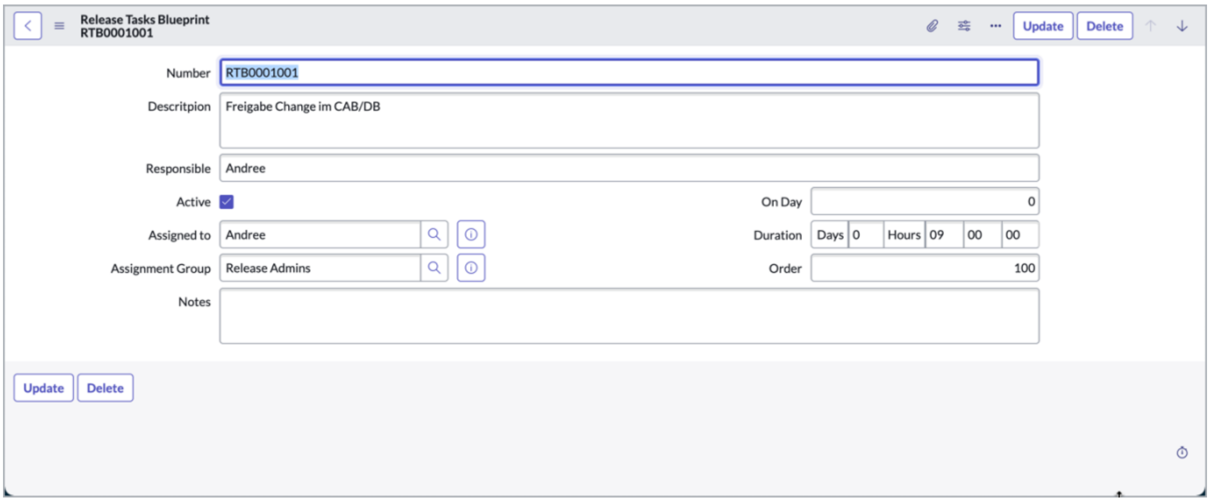


When you have done your changes, click "Save" in the upper-right corner of the window. Click "Preview" to open the Form view for the table on a new tab in App Engine Studio (see next screenshot below).

Info:
Try to arrange the fields on the form in the way shown in the screenshot below (it is not mandatory to arrange exactly in this layout, but helps to learn about the Form Builder).



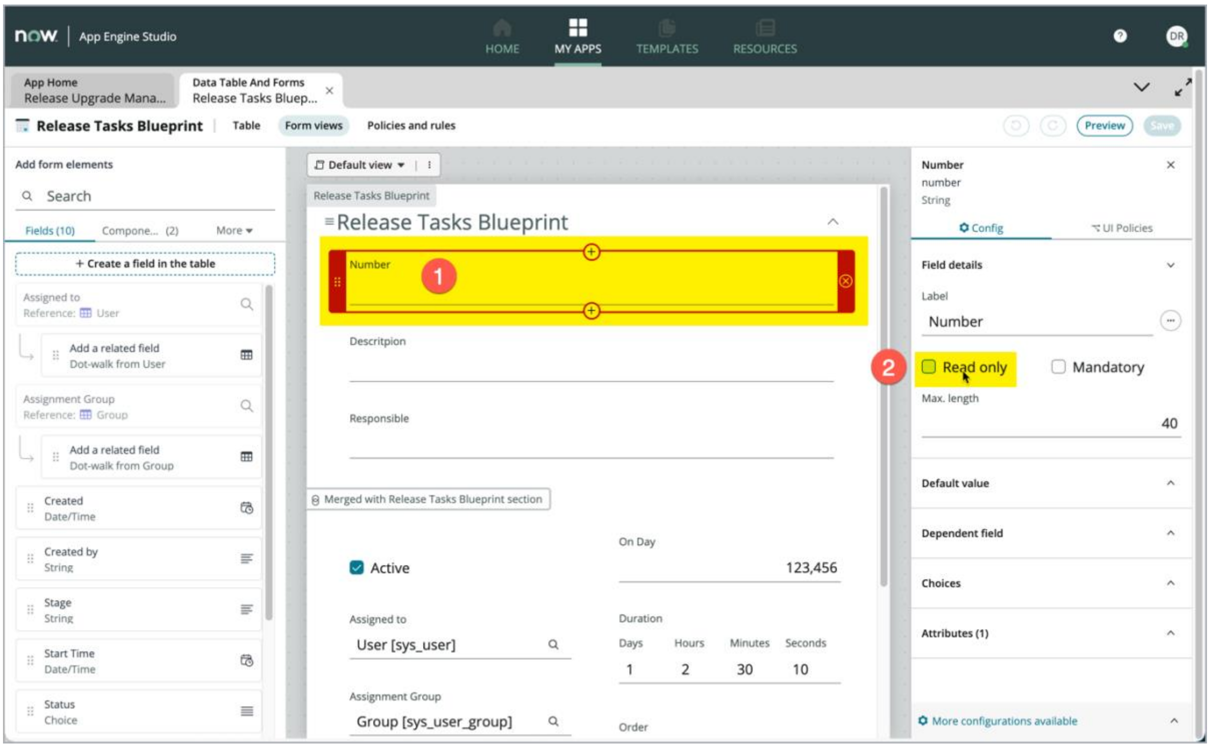
You can also click "**Open form in Platform**" in the upper-right corner to open the form in the Experience shown below (which is called the **Platform UI**):



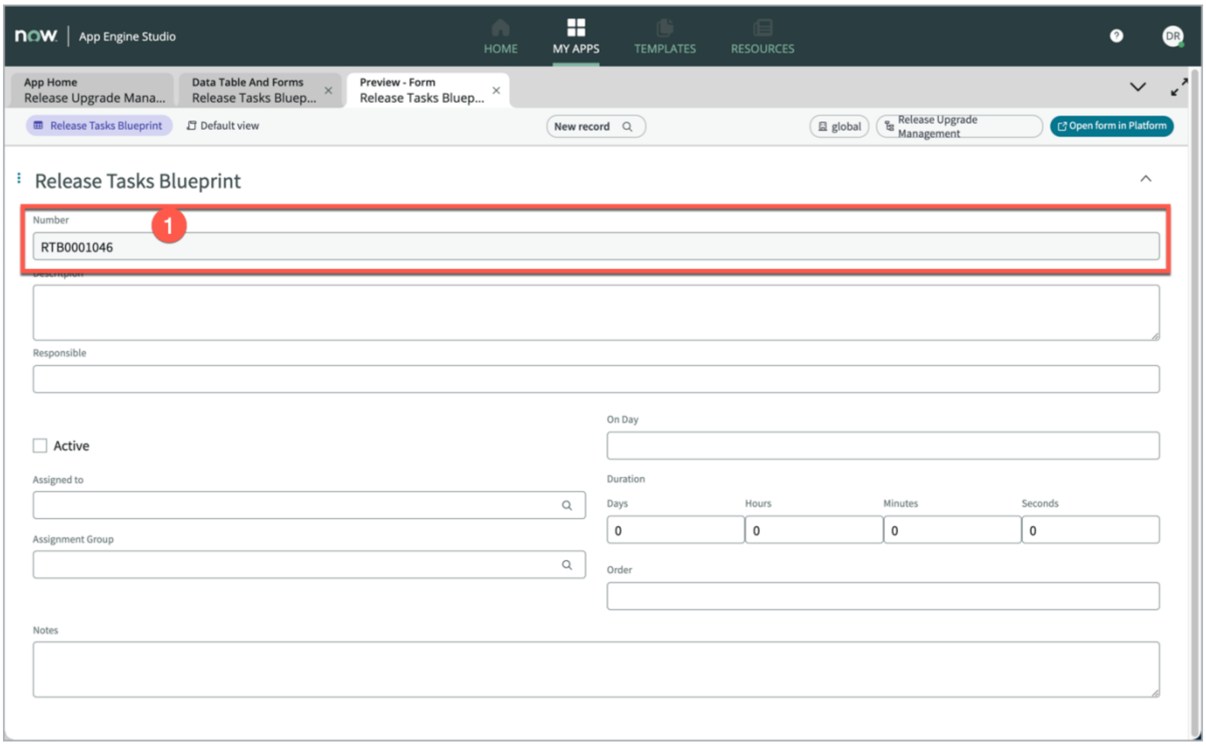
Navigate back to App Engine Studio and close the extra Tab in App Engine Studio, so that you now are back in the Form Builder (see below):

Select the “**Number**” field (1) on the Form and **mark the “Read only”** (2) Checkbox in the right-hand panel.

Click “**Save**” in the upper-right corner and **re-open the Preview**.
You will see, that the “Number” field now cannot be edited anymore.



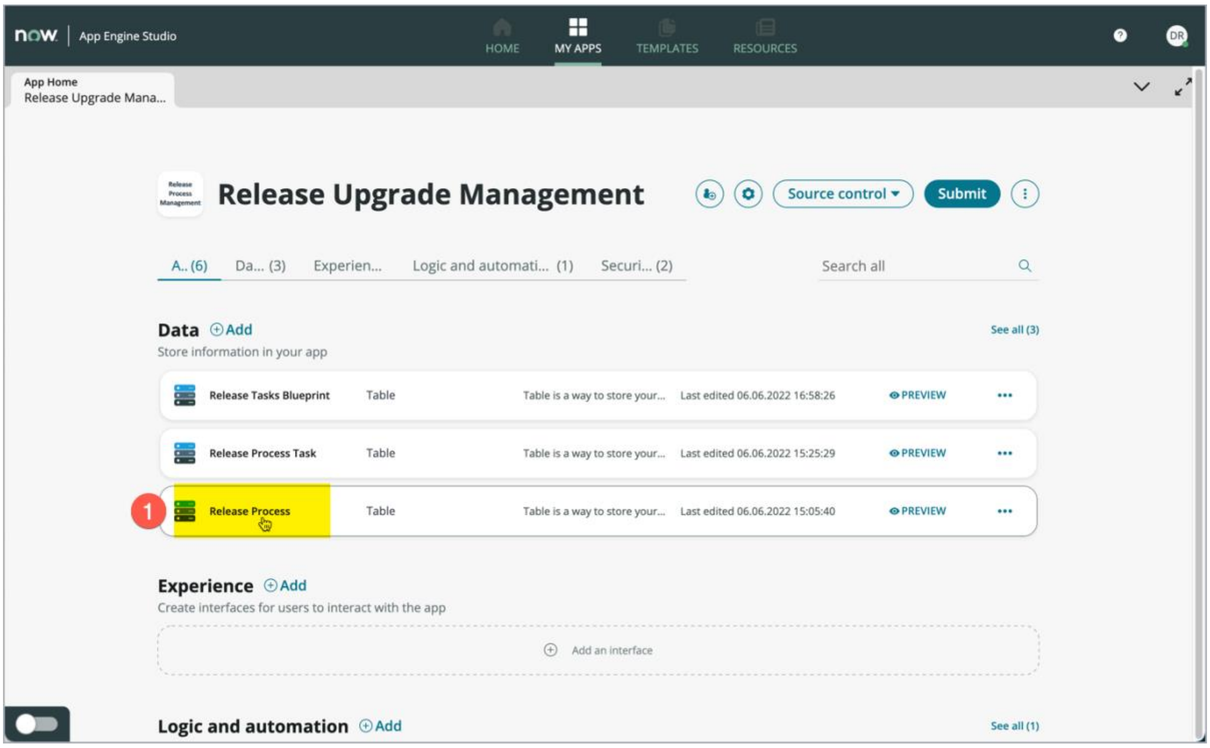
The “Number” field is now shown “grayed-out” (1), which means that you cannot edit it anymore:



Congratulations, you have configured the Form layout for the “Release Task Blueprint” table.

3.1.2 Layout for the "Release Process" table.

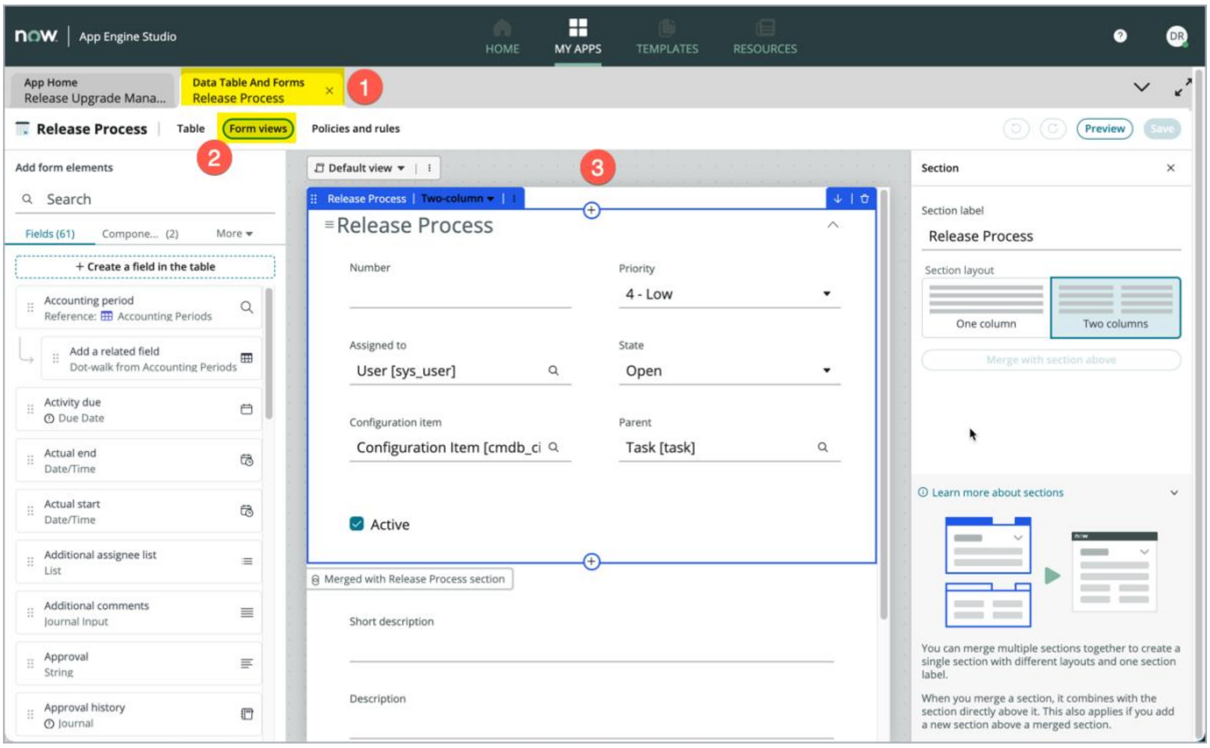
From the dashboard of your App in App Engine Studio, open Table Builder for the "Release Process" table by clicking the table name in the Data section.



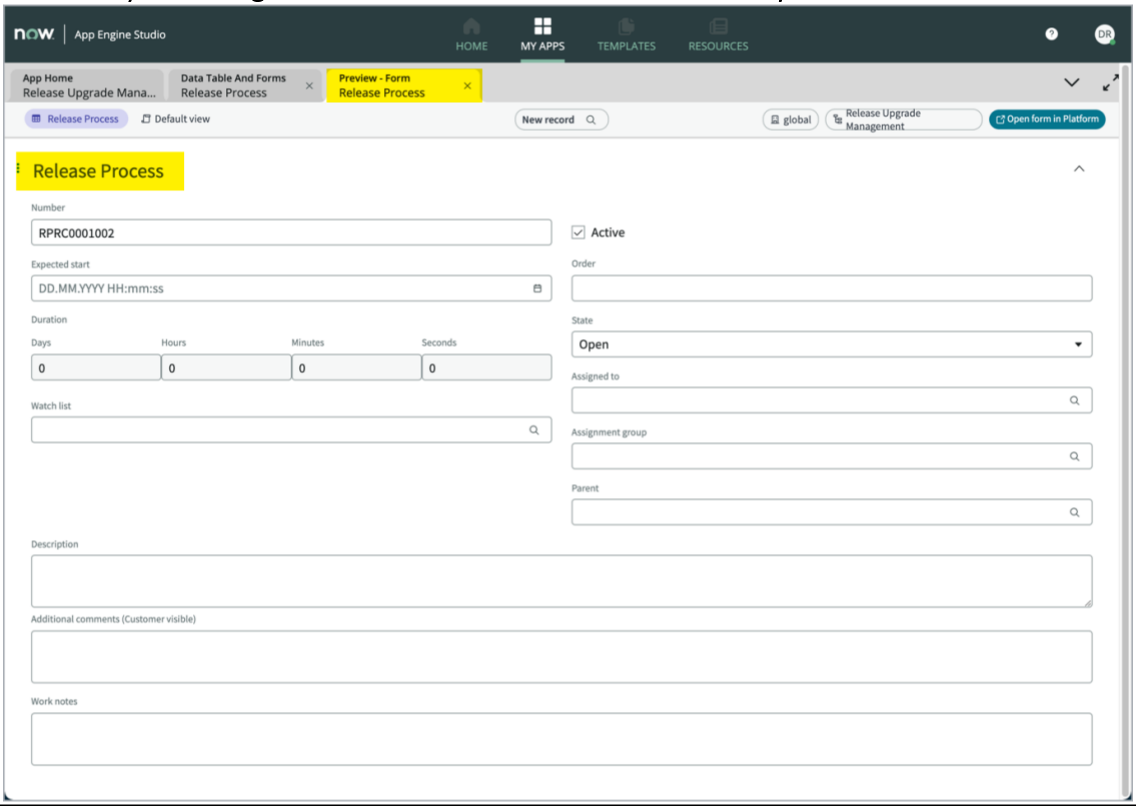
a) Configure the FORM layout

In Table Builder (1) switch to the **“Form views” Tab** (2) and review the layout, that ServiceNow created automatically for the table (which in fact is based on the default layout of the Task table – the table we used to extend for the “Release Process” table).

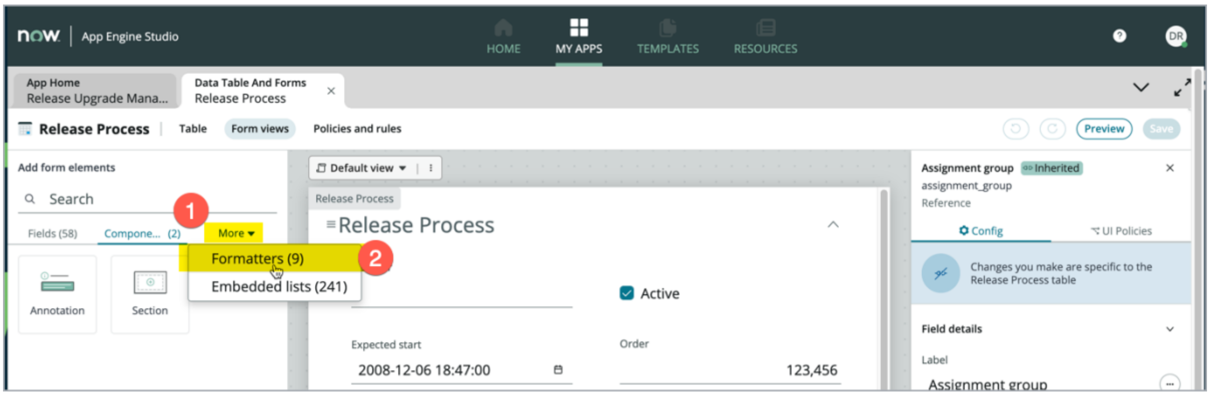
There are **several fields available**, and some fields already on the form does not make sense to our use case.



Review the Form layout that I created, which seems to make sense for me – see screenshot below. Try to arrange the fields on the form in the same way.

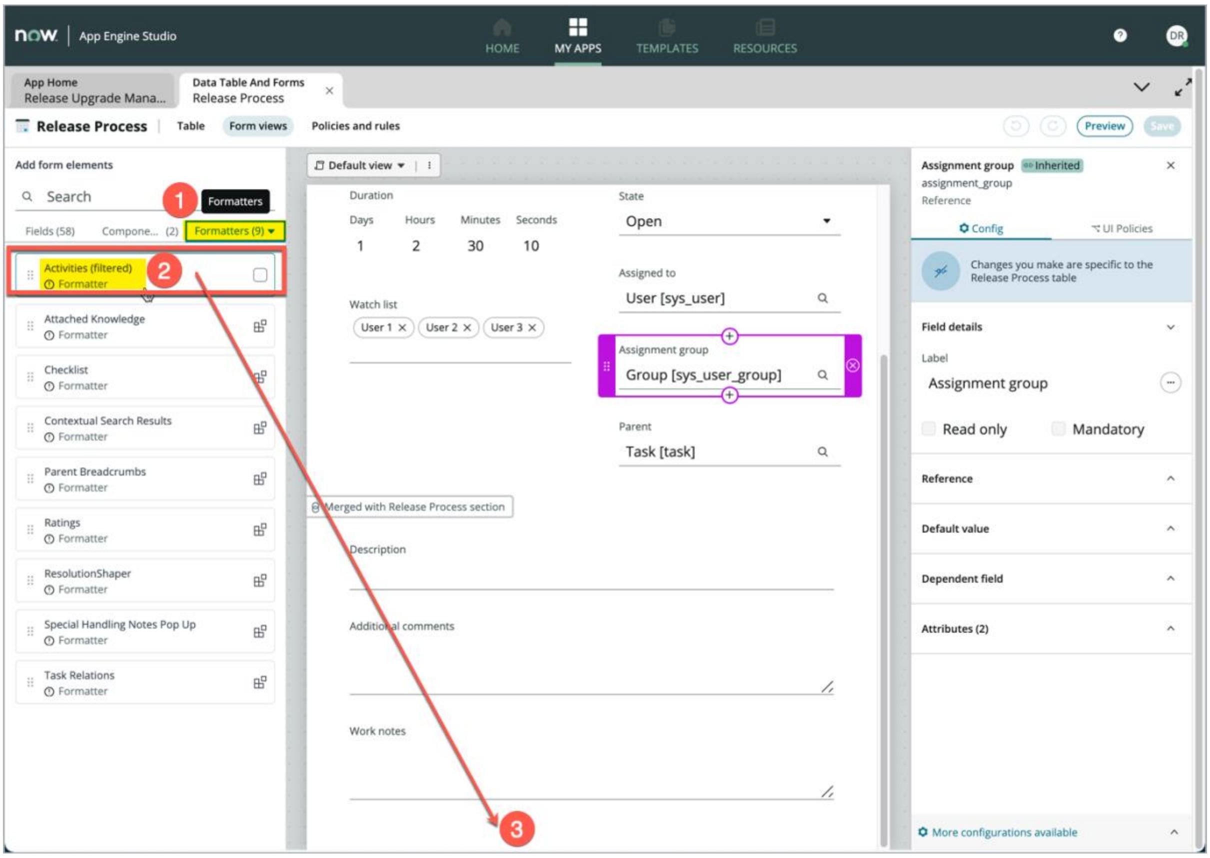


Back in Form Builder, switch to the "**Formatters**" Tab (2) in the right panel (1)



On the "Formatters" Tab (1) **drag the "Activities (filtered)" Formatter** (2) to the very bottom of the Form (3).

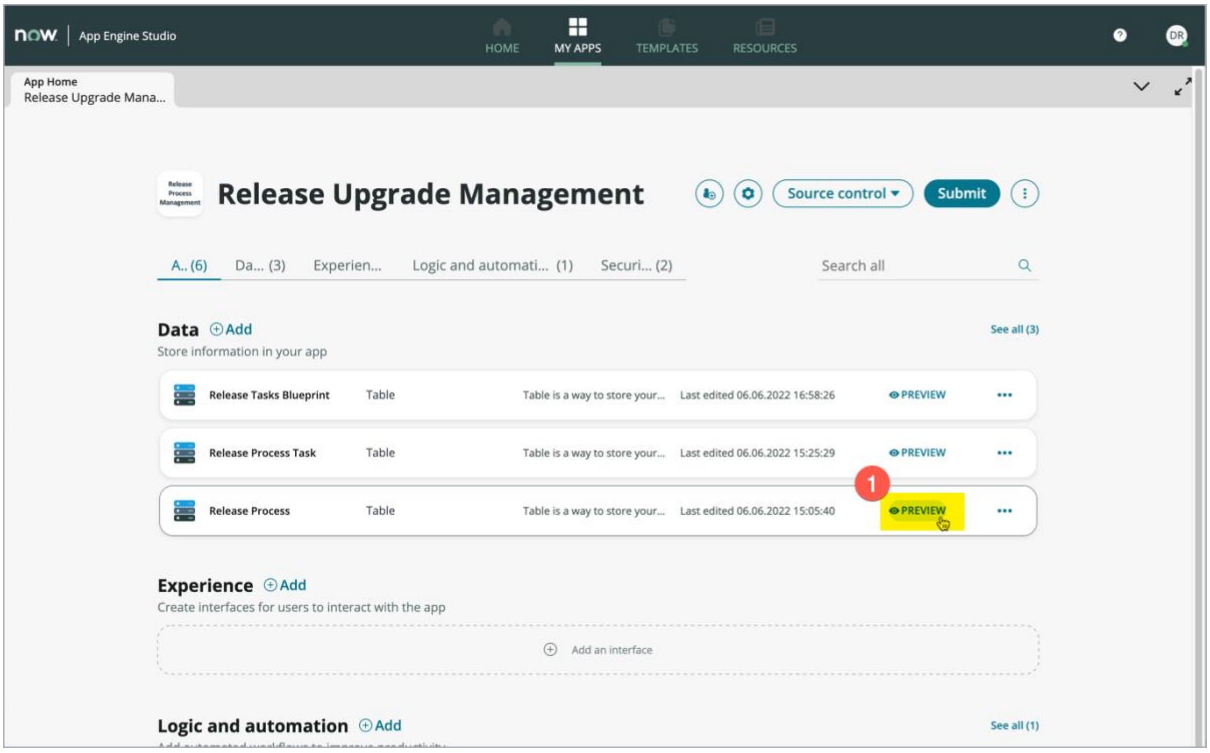
Click "**Preview**" (upper-right) again.



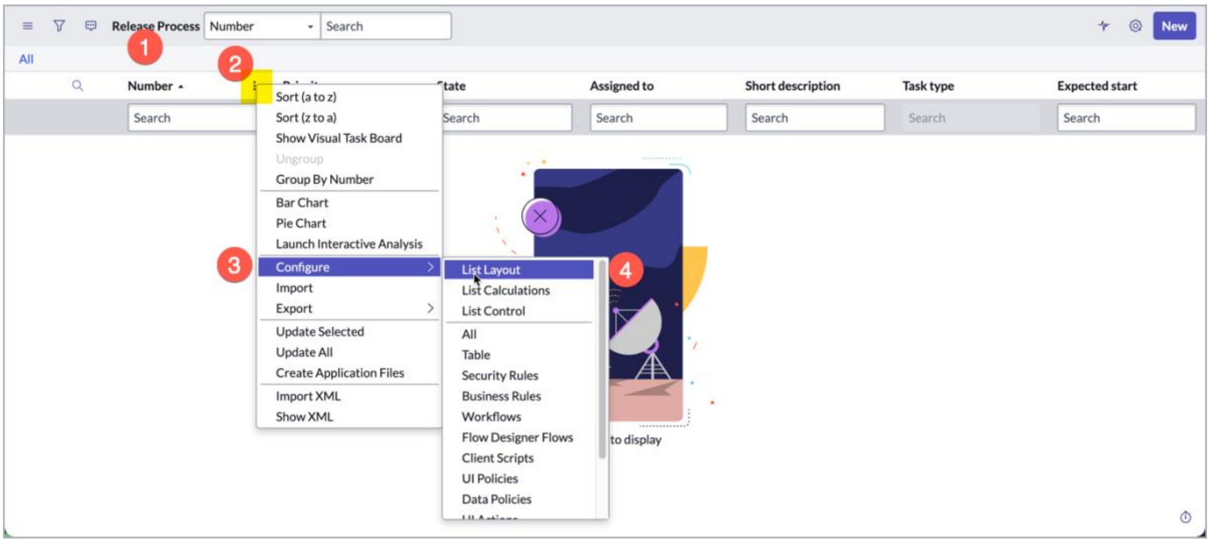
Close the "Table Builder" Tab here in App Engine Studio and navigate back to the App Dashboard.

From the dashboard, click “**Preview**” (1) on the “**Release Process**” table in the “**Data**” section.

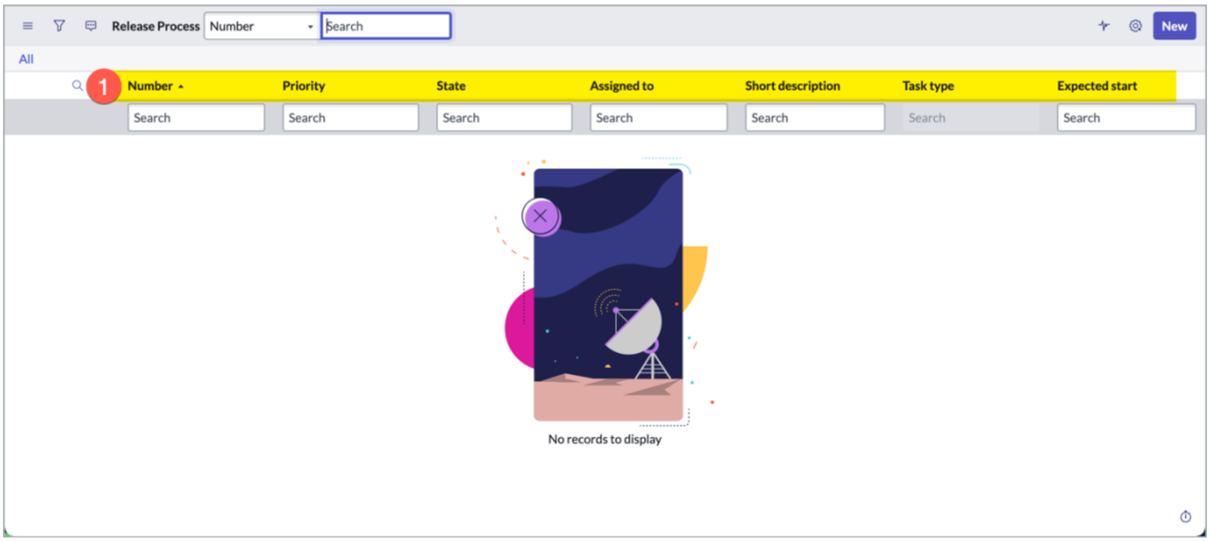
This will open the table in List View in a new Tab in your Web Browser.



Configure the List Layout.



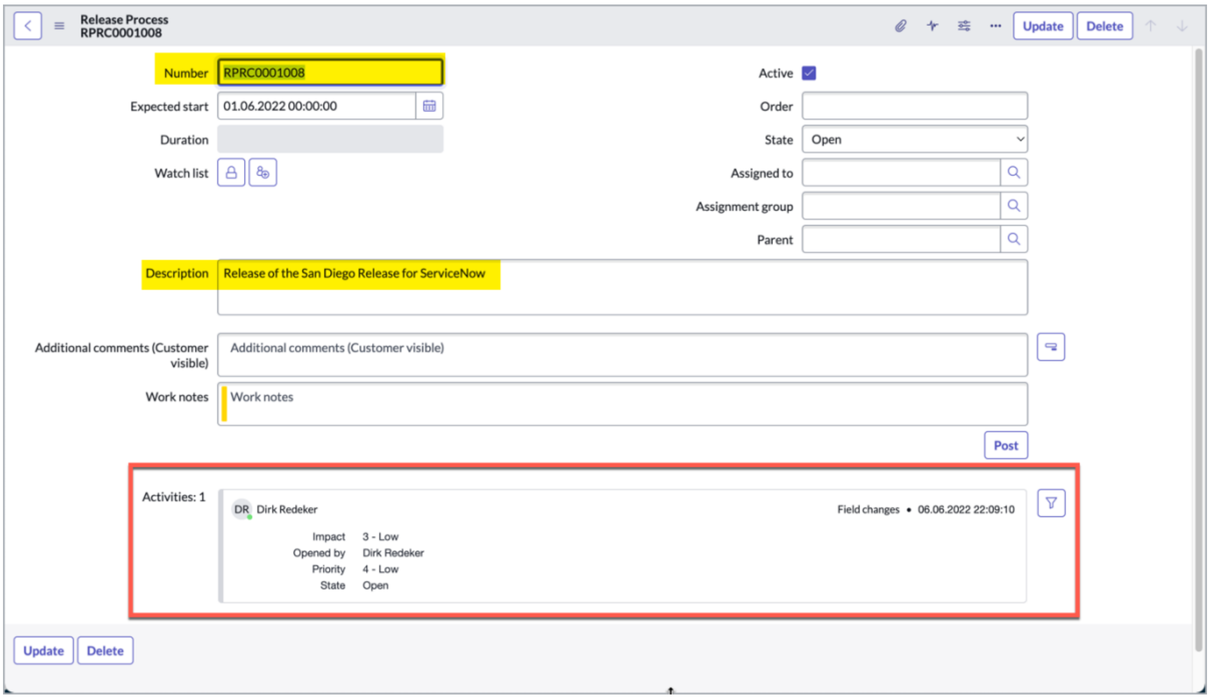
Info:
Just for training puporses, try to make the List layout look (1) same or like the one shown in the screenshot below.



Click “**New**” in the upper-right corner and **capture a new “Release Process” Record**.

Fill in the form and right-click the header to “Save”.

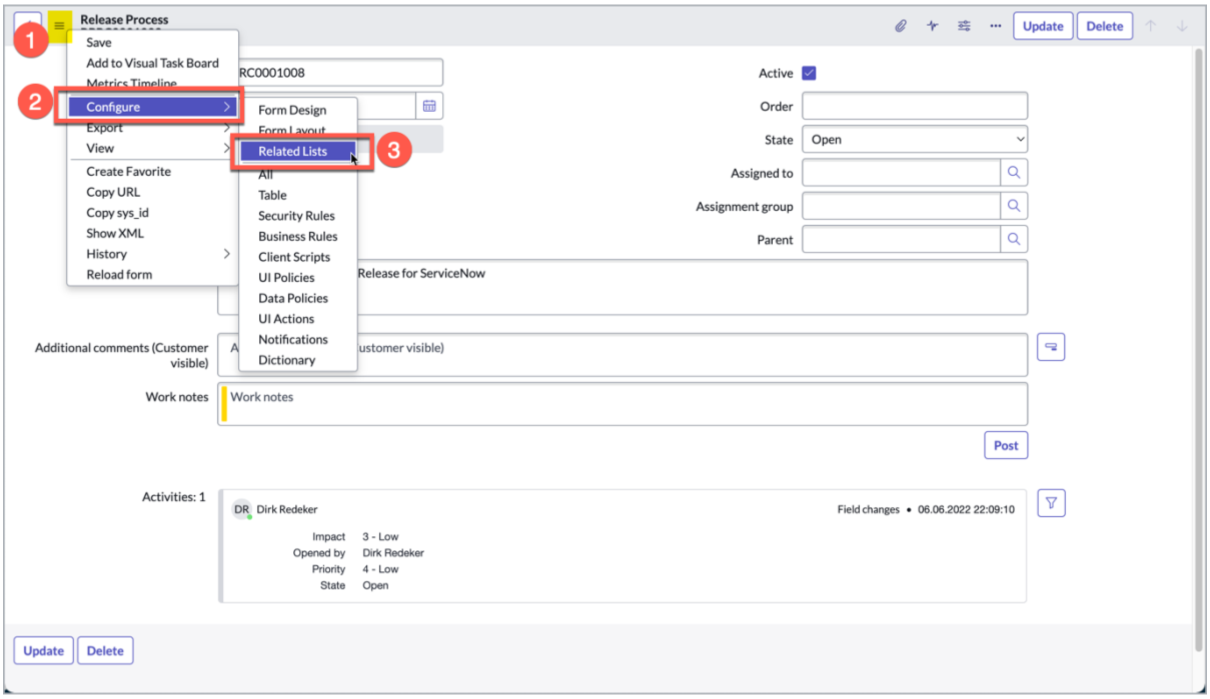
Note:
The “**Activity Stream**” which is shown in the bottom of the form. It logs the changes done on the record. This gives a historical view on the data without and further coding.



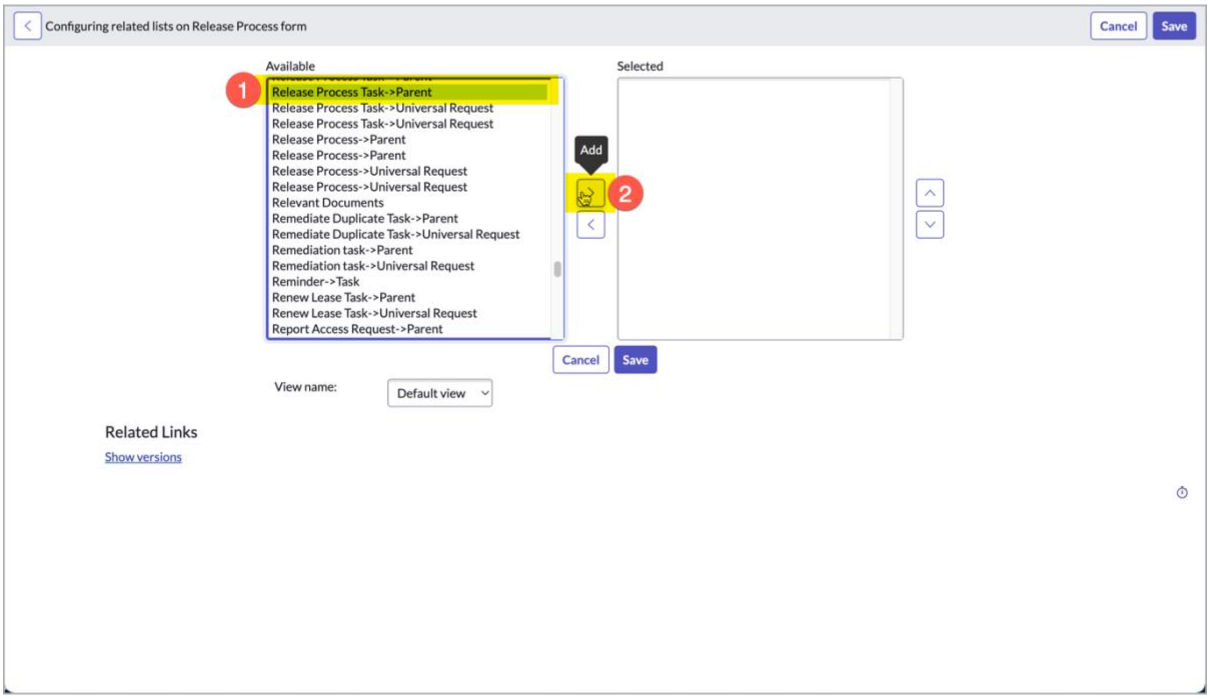
Finally, let's add a (so called) Related List to the form. The Related List will show all "Release Process Tasks", that are associated/linked to this one specific "Release Process".

Each single Release Process record can now have any number of Tasks associated to it individually.

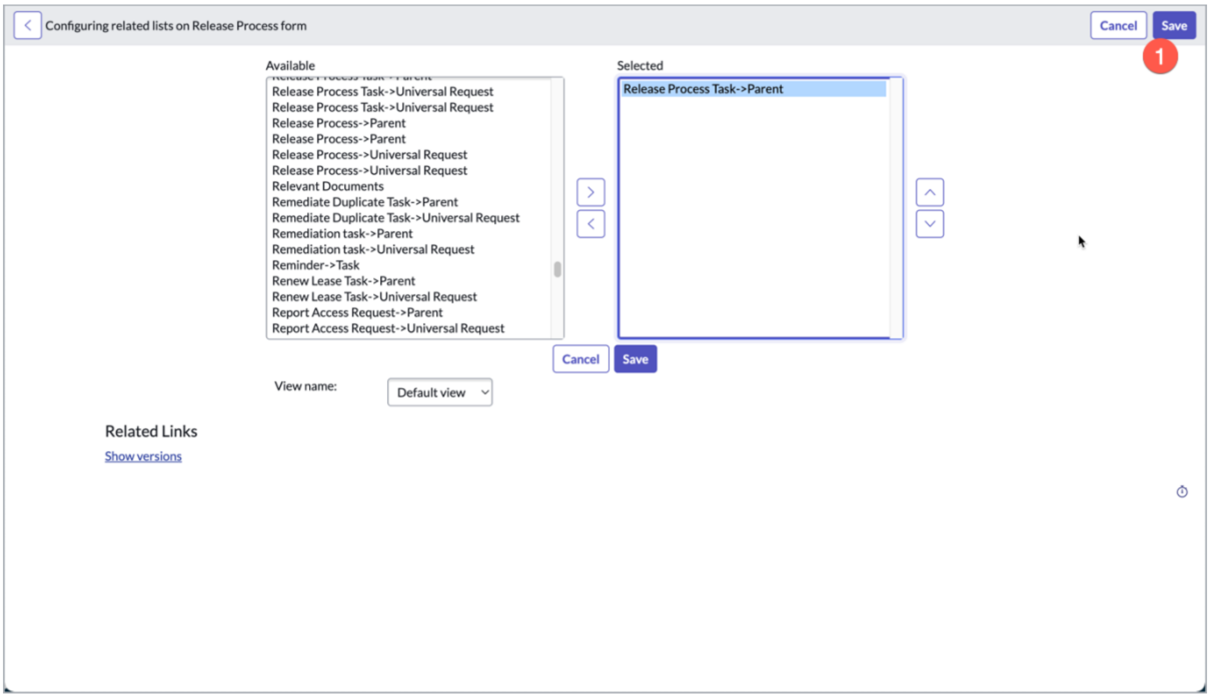
Click the Form menu (1) (hamburger menu in the top-left corner) and select "Configure" (2) and then "Related Lists" (3).



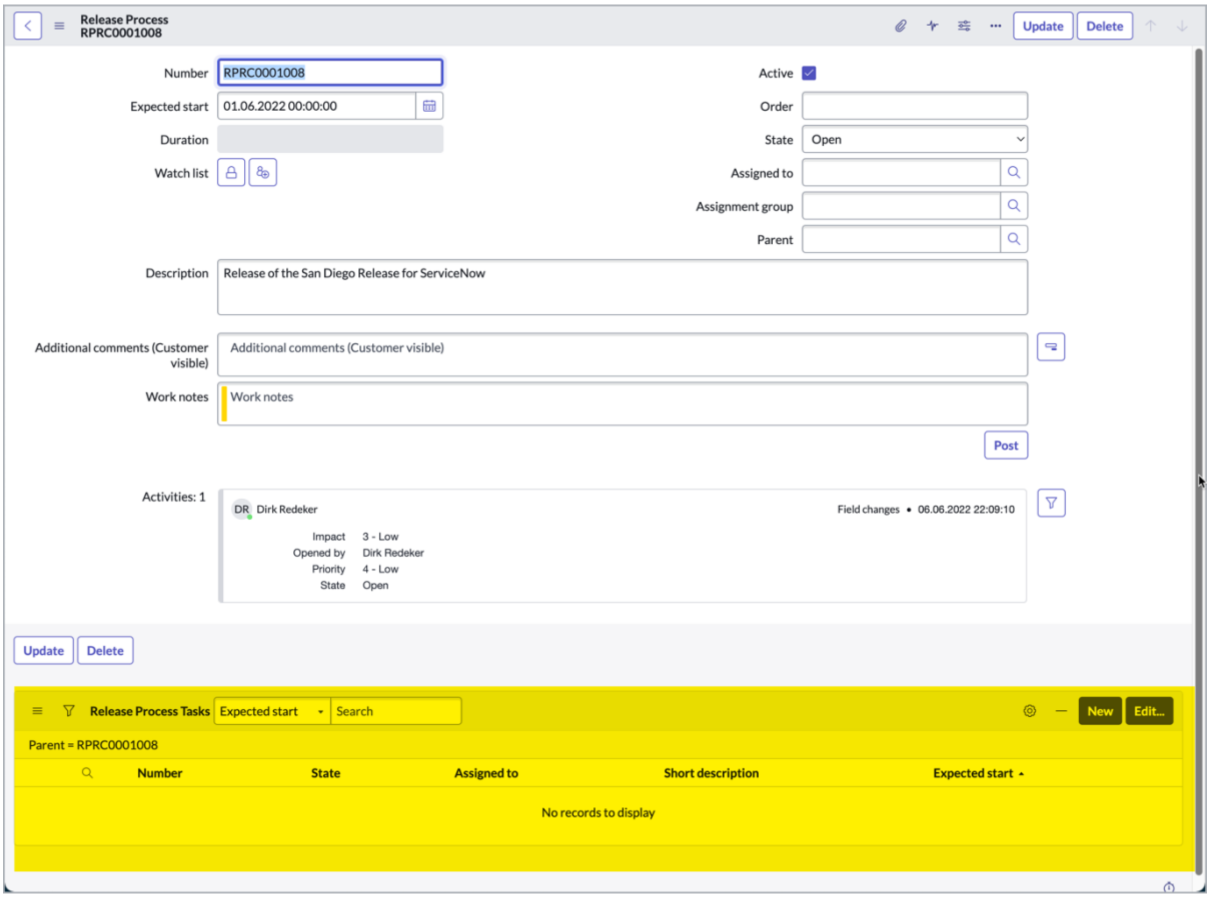
Find the entry “**Release Process Task -> Parent**” in the left list, select it (1) and move it to the right list using the “**Add**” button (2)



Click “**Save**” (1) to close the Dialog and make the changes permanent.



Review the Form, that there is now a list shown at the bottom of the Form.



Finally, lets Configure the "Release Process Task" table layout

3.1.3 Layout for the "Release Process Task" table

We do not need a bunch of screenshots again.
You now learned about how to configure List and Form layouts.
Take a moment to design the List and Form layout for the "Release Proces Task" table

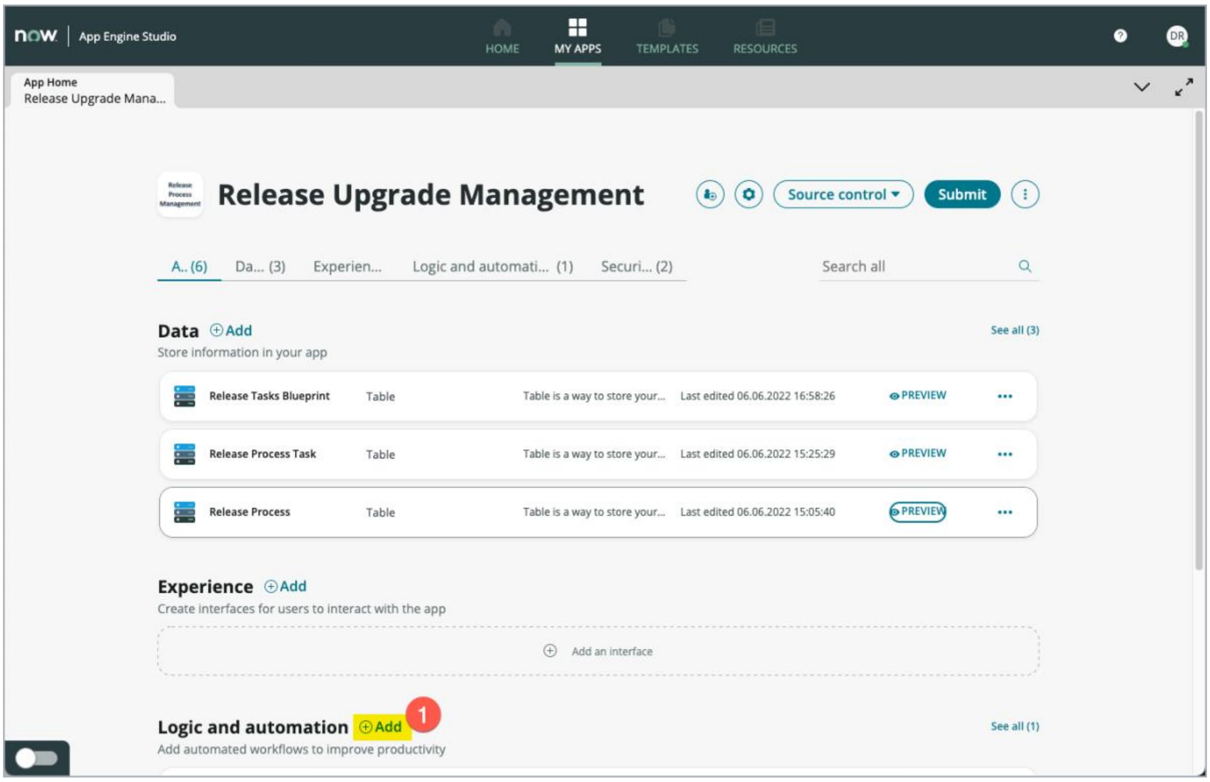
Note :
This step in the exercise is optional and you decide about the layout and outcome her.
Have fun!

4 Add automation to your App

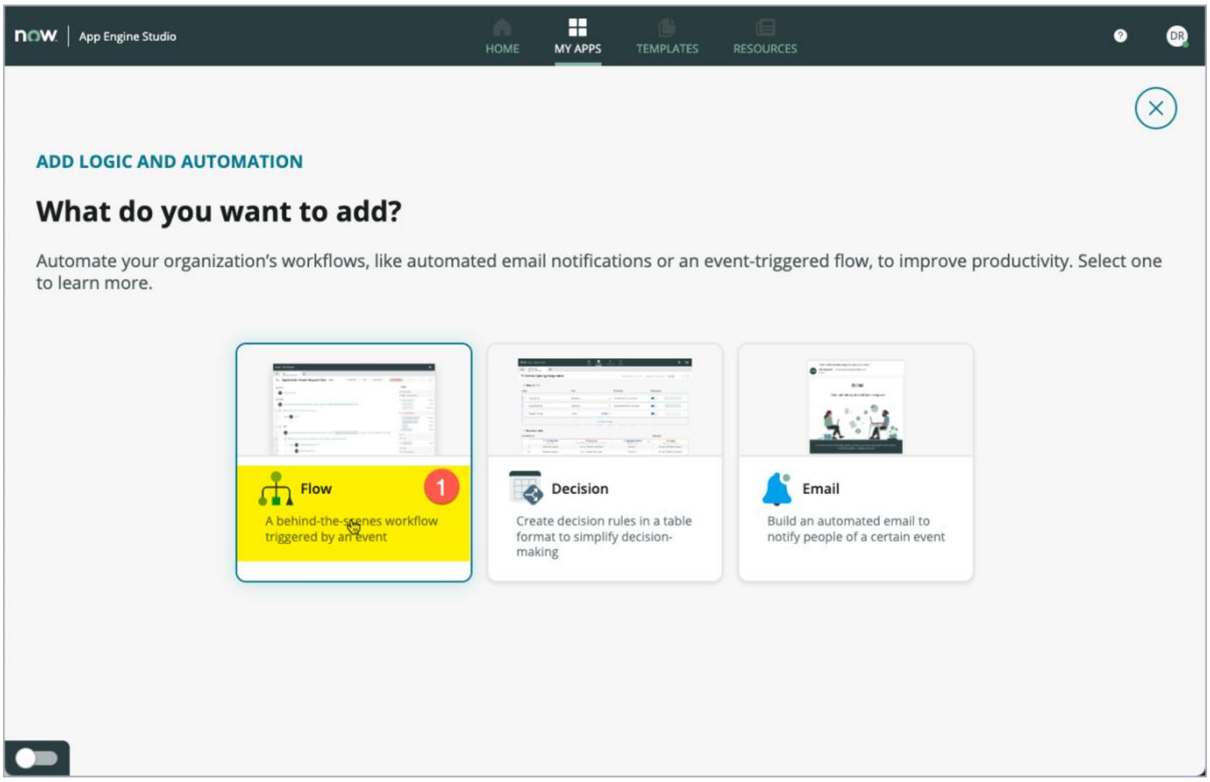
4.1 Exercise #4 – Create Automation to copy the Task Blueprint

Start from the dashboard of your App in App Engine Studio

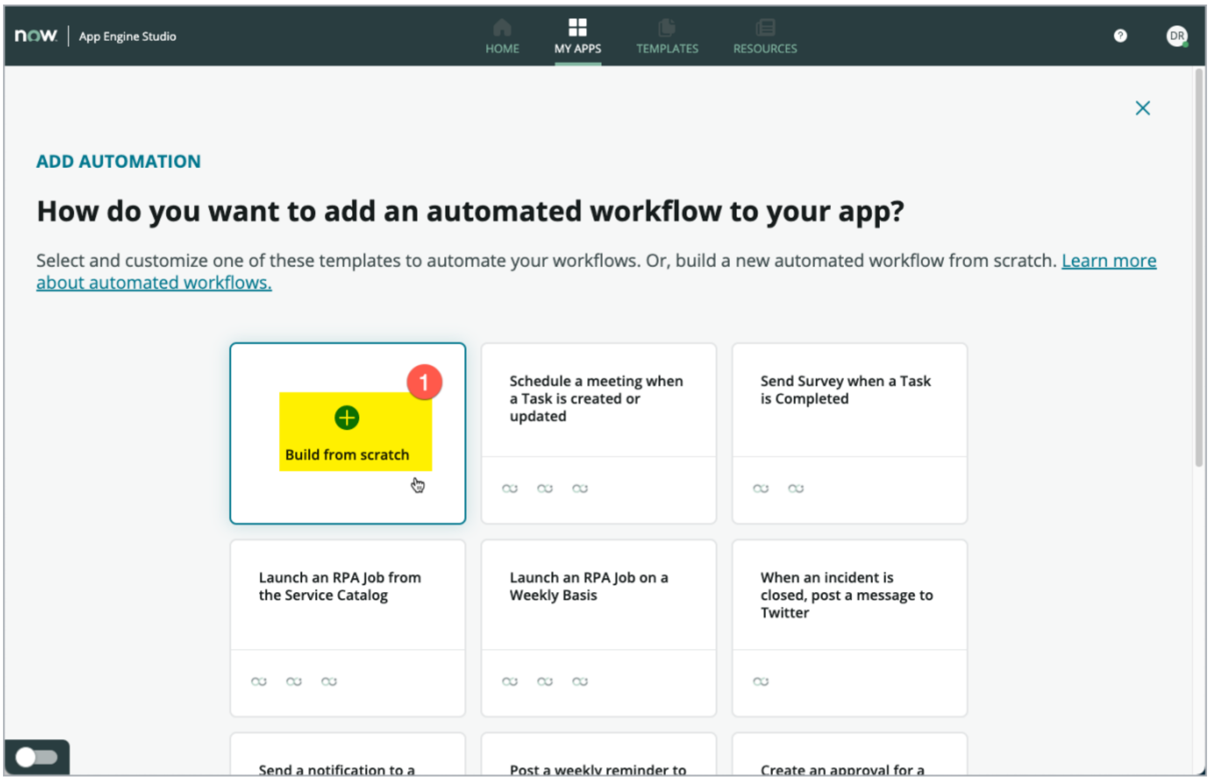
Click "**Add**" in the "Logic and automation" section (1)



Select the left-hand card “**Flow**” (1).



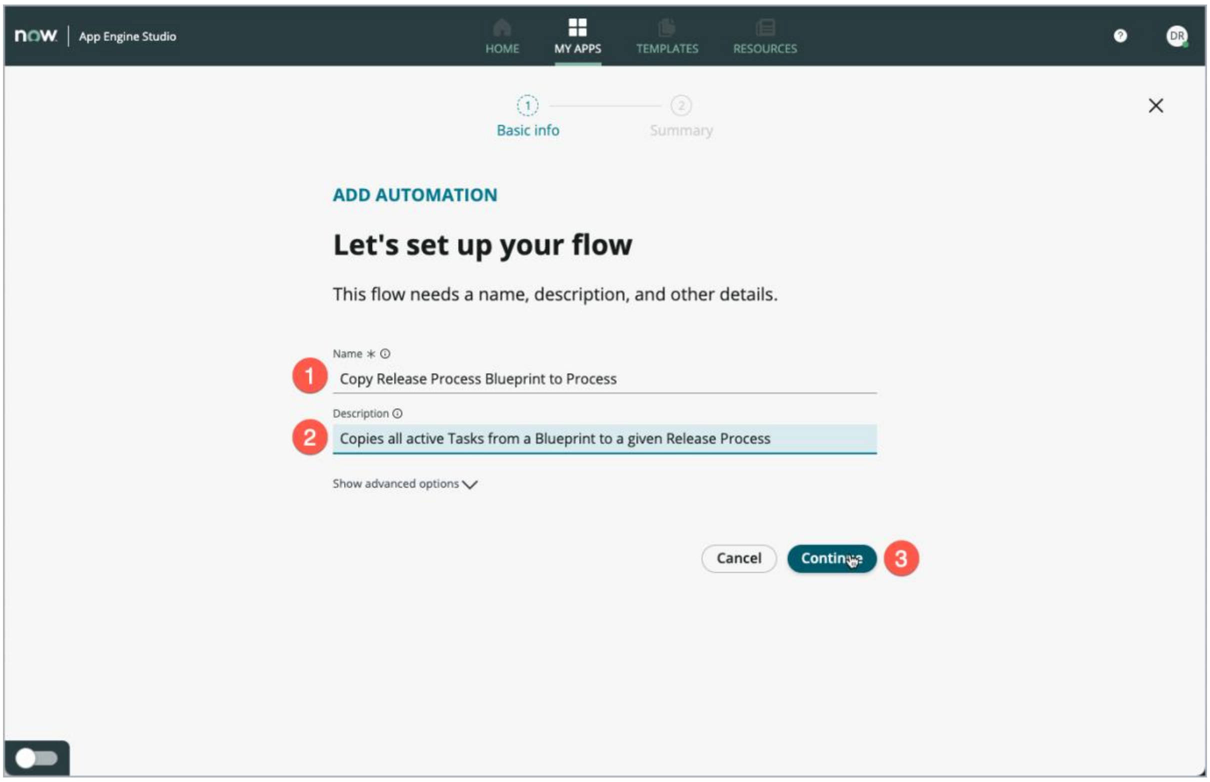
Click “**Build from scratch**” (1)

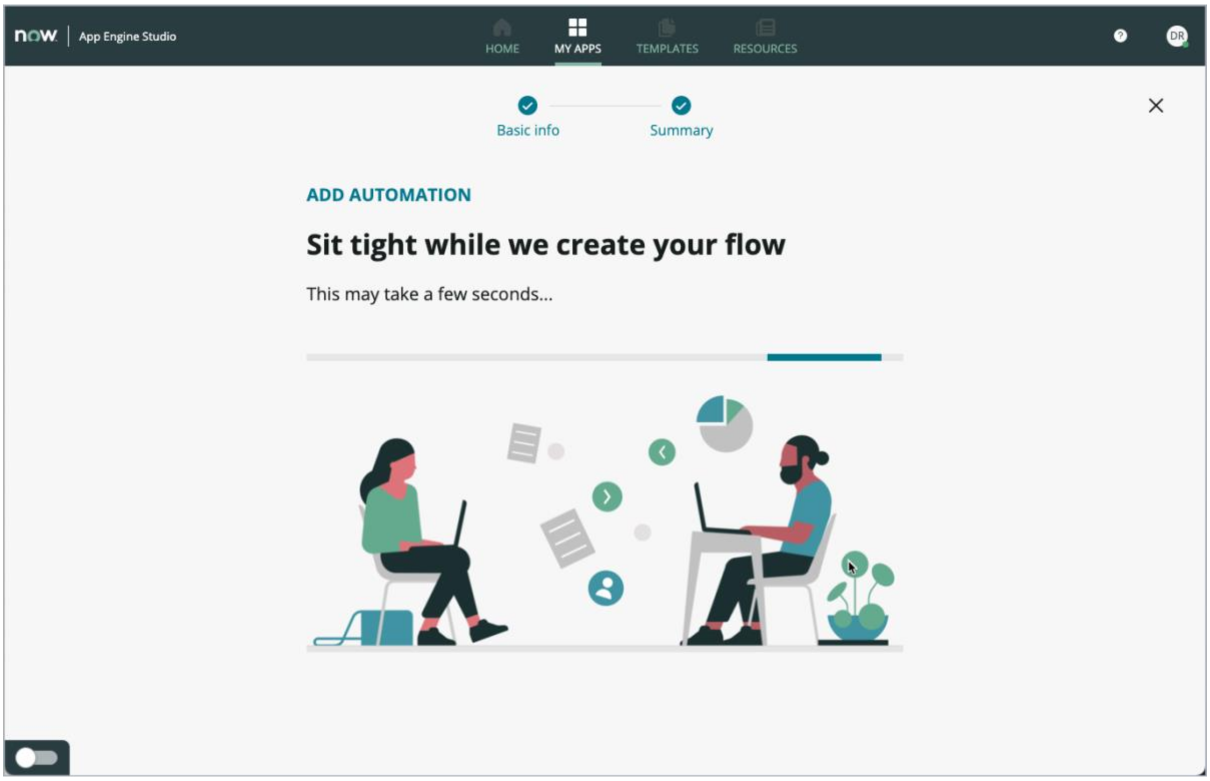


Enter a **meaningful (and unique) name** for your new Flow in the "**Name**" (1) field. Optionally, capture a short "Description" below (2).

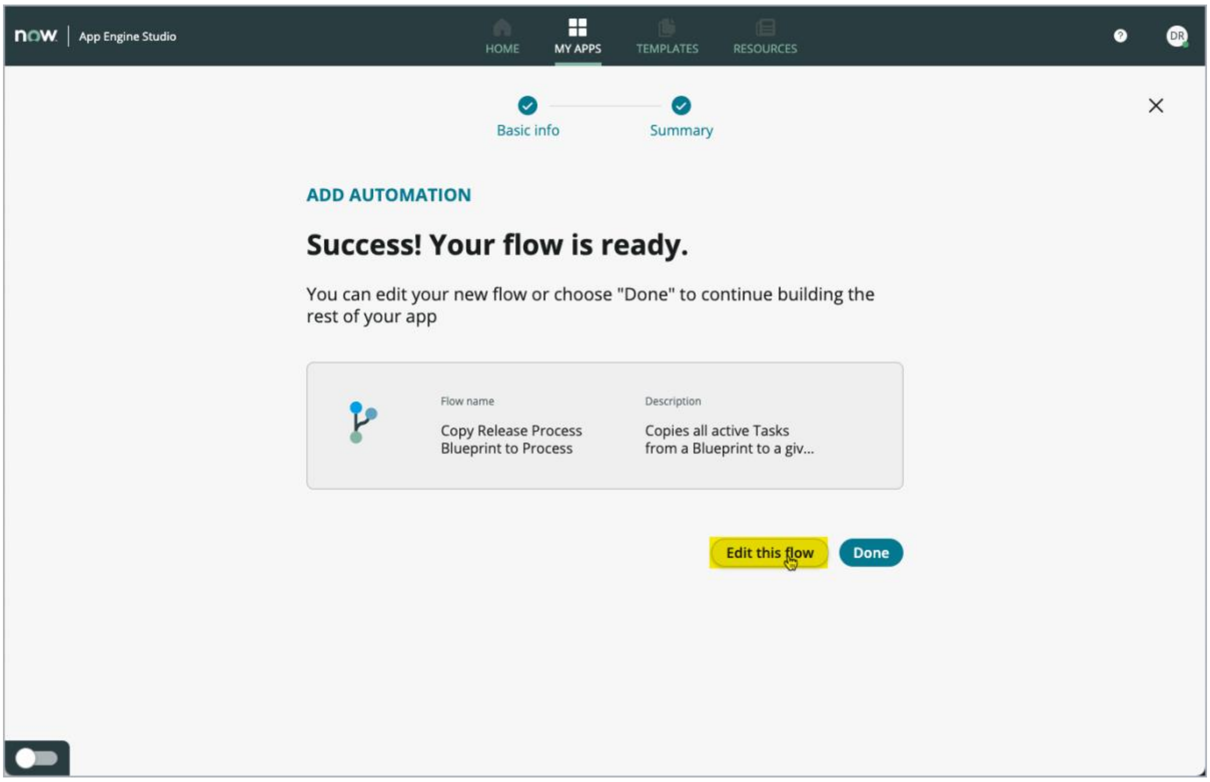
Click "**Continue**" (3) to move on.

Note:
It's a good idea, to create and define a "naming schema" for your organization, so that the overview of Flows in your instance stays clean and understandable – even if you can search and sort by Scope.



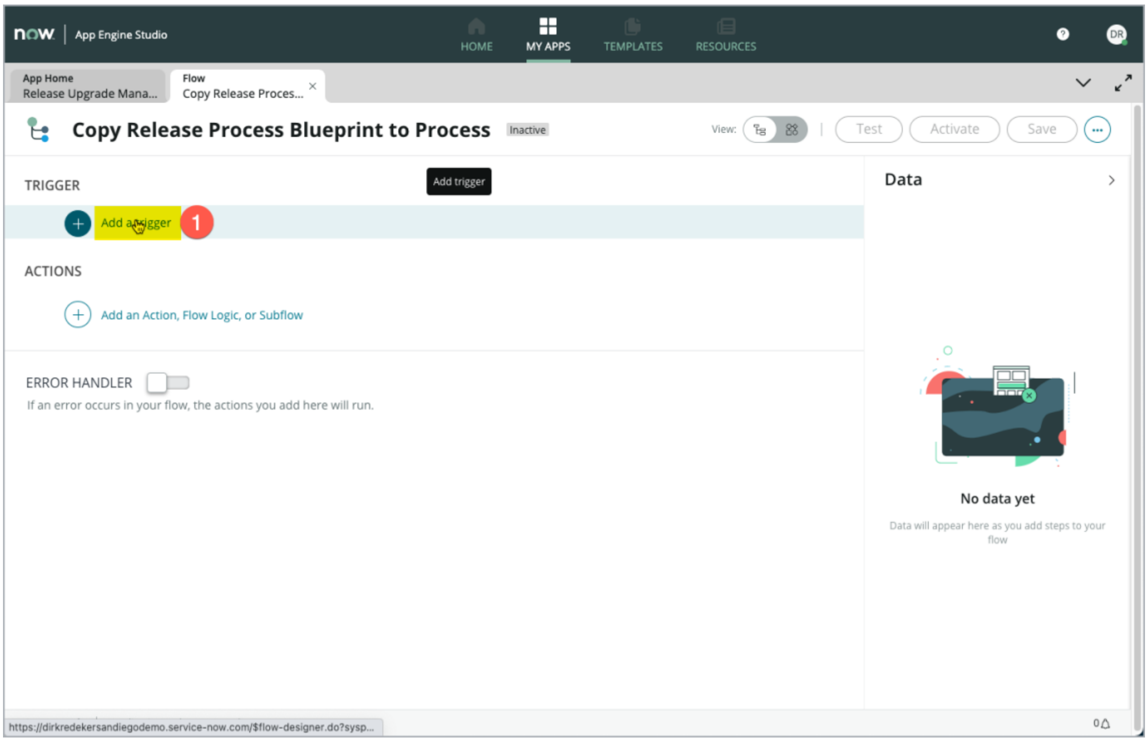


As soon as the new Flow is created click “**Edit this flow**” (1) to move on.



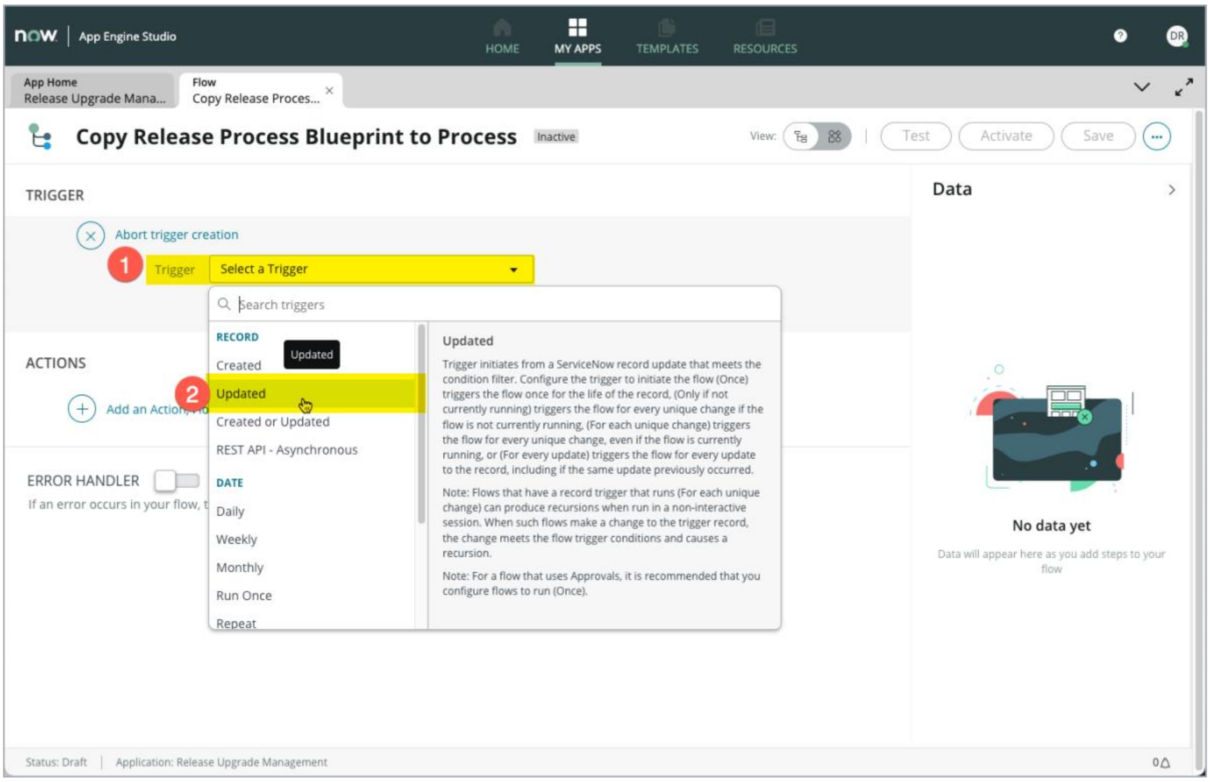
This opens Flow Designer. Wait a few seconds, while Flow Designer is loading, and the screen is ready for editing.

Click “**Add a trigger**” (1).



The Dropdown for the "**Trigger**" type field (1) opens automatically.

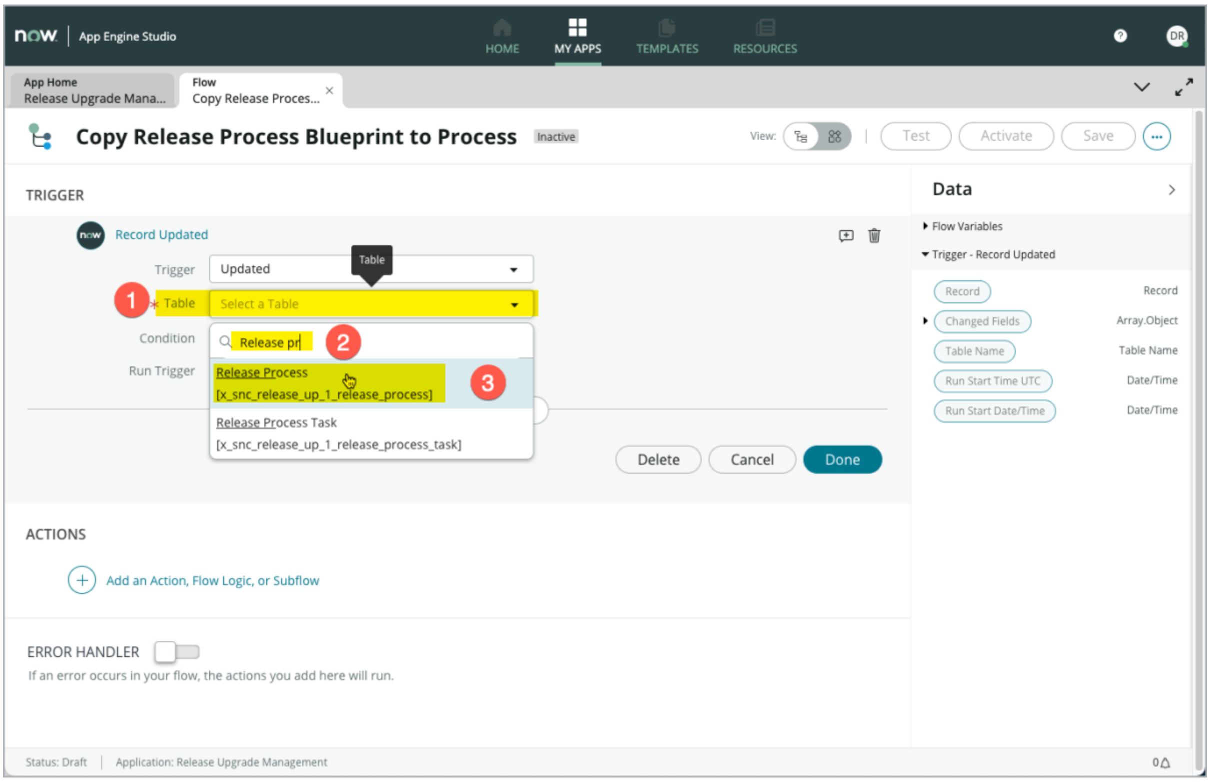
Select "**Updated**" (2) from the List.



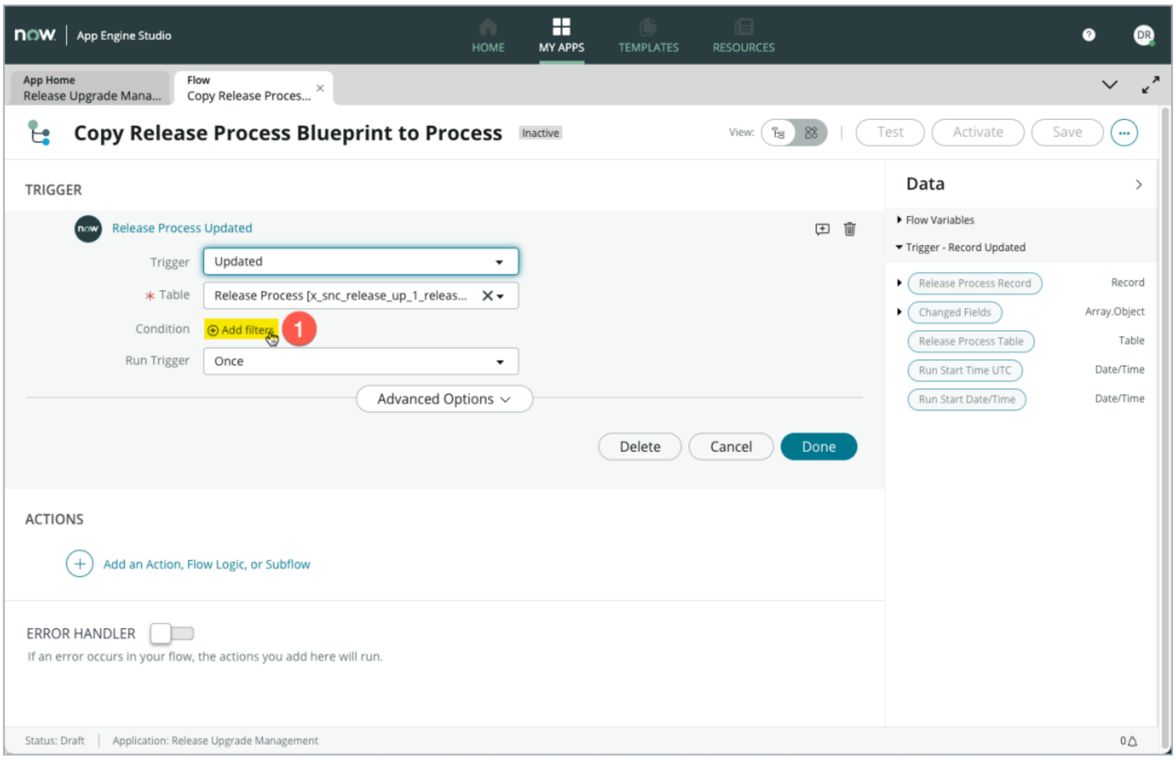
Note: Flow designer does "heavy caching"!
This means, if your table does NOT appear in the list here, Close Flow Designer and also App Engine Studio. Re-Open both to refresh the cache – this should make new tables also appear in the list.

Click the "**Table**" field (1) and **type "Release pr"** in the filter box (2). This limits the number of shown tables in the Dropdown list, so that you easily can find your table.

Select the "**Release Process**" table from the List (3).



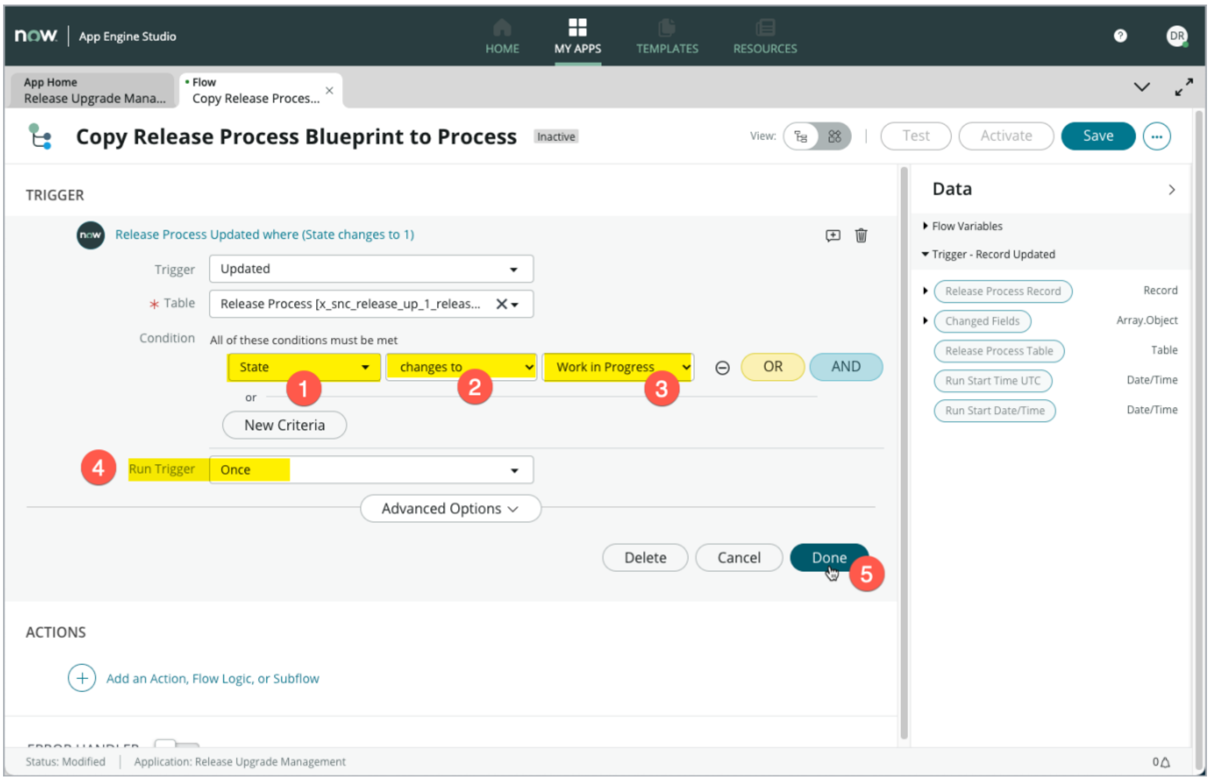
Click "**Add filters**" (1)



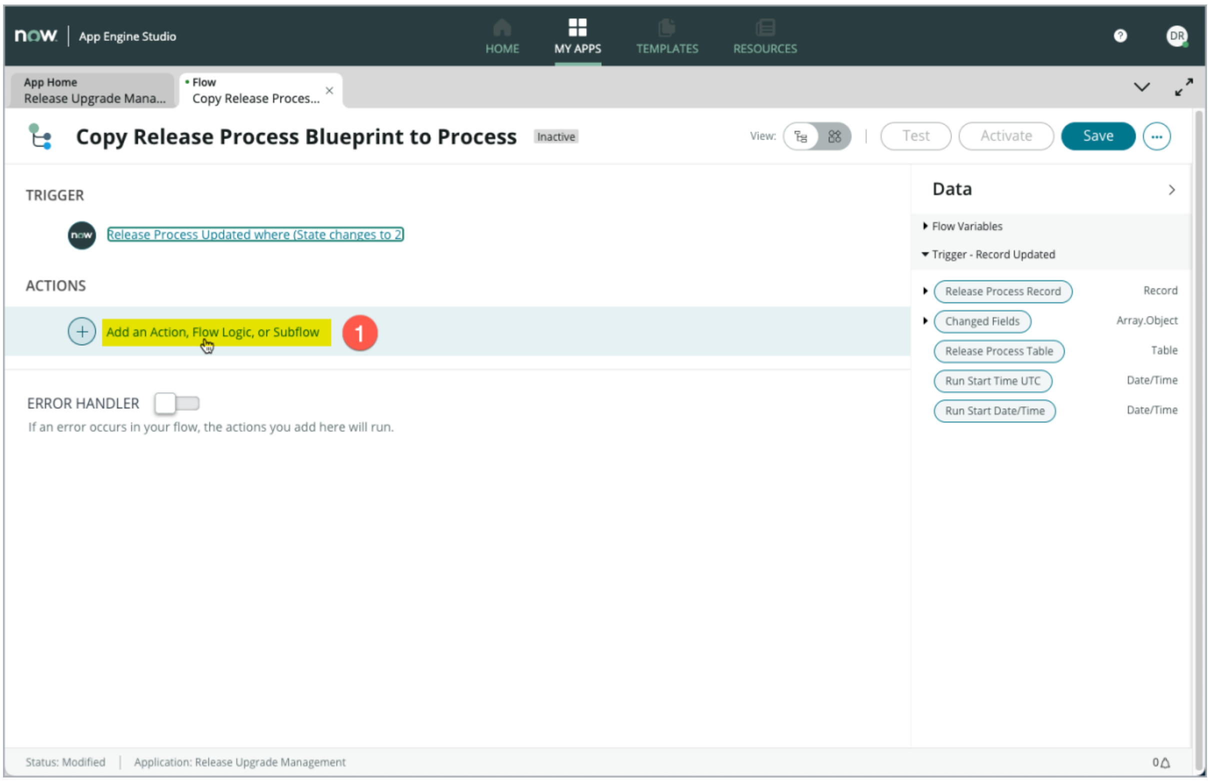
Select “**State**” in the first dropdown (1) to select the table field for the condition.
Select “changes to” in the second dropdown (2) and select “Work in Progress” from the third dropdown (3).

Do **NOT change** the “Run Trigger” field (4). Leave this to the value “Once”.

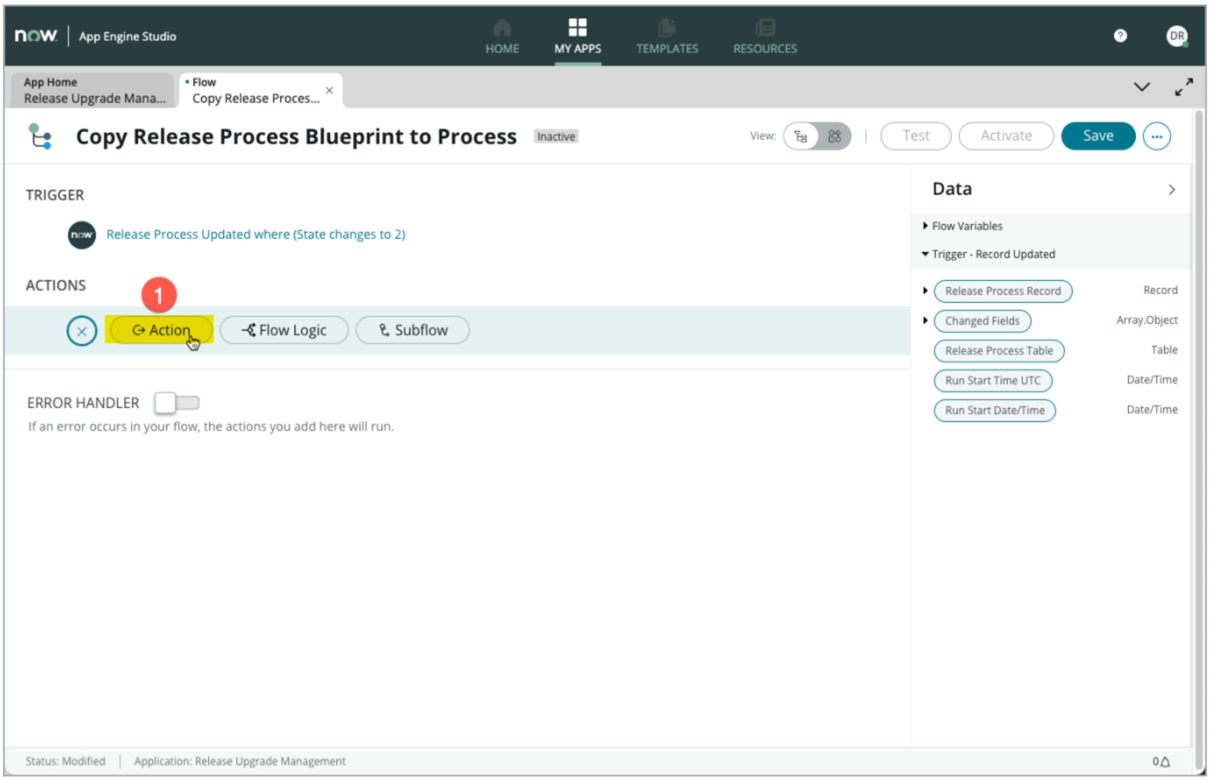
Click “Done” (5) when finished.



In the **"Actions" section** of Flow Designer, click "**Add an Action, Flow Logic, or Subflow**"



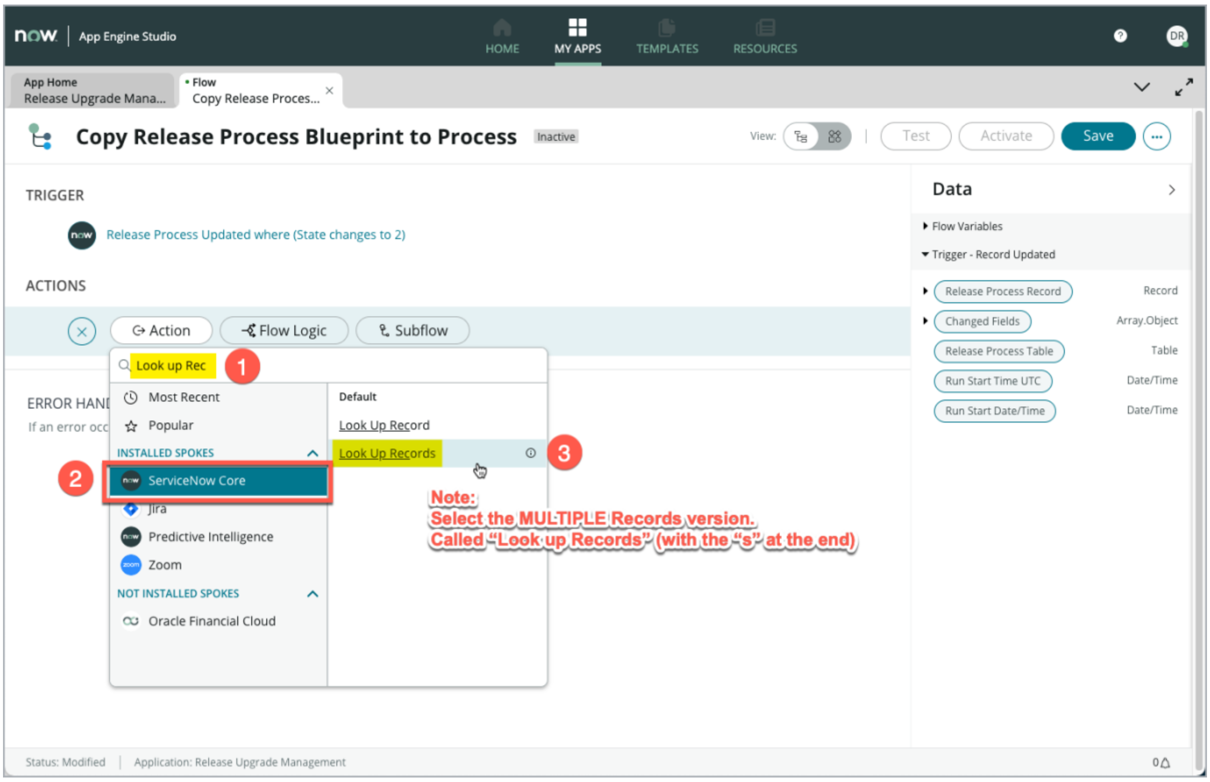
Click "**Action**" (1) to add an Action to your Flow.



In the filter row (1) type "Look up Rec" to limit the number of shown Actions in the dropdown. Click "**ServiceNow Core**" (2) to further limit Action from that Spoke.

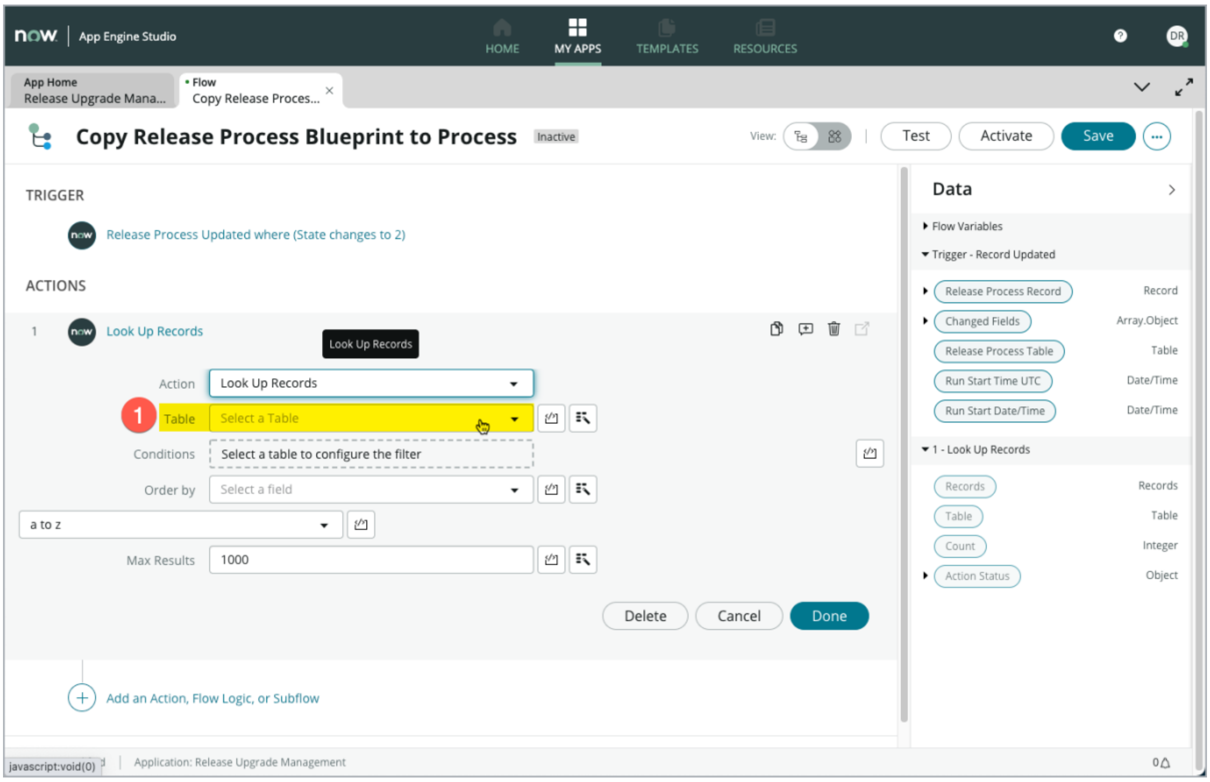
Select the "**Look Up Records**" Action (3).

Note:
You need to select the Action to **look up MULTIPLE Records**. This is the one, with the "s" at the and in the Name "Look Up Record**s**")

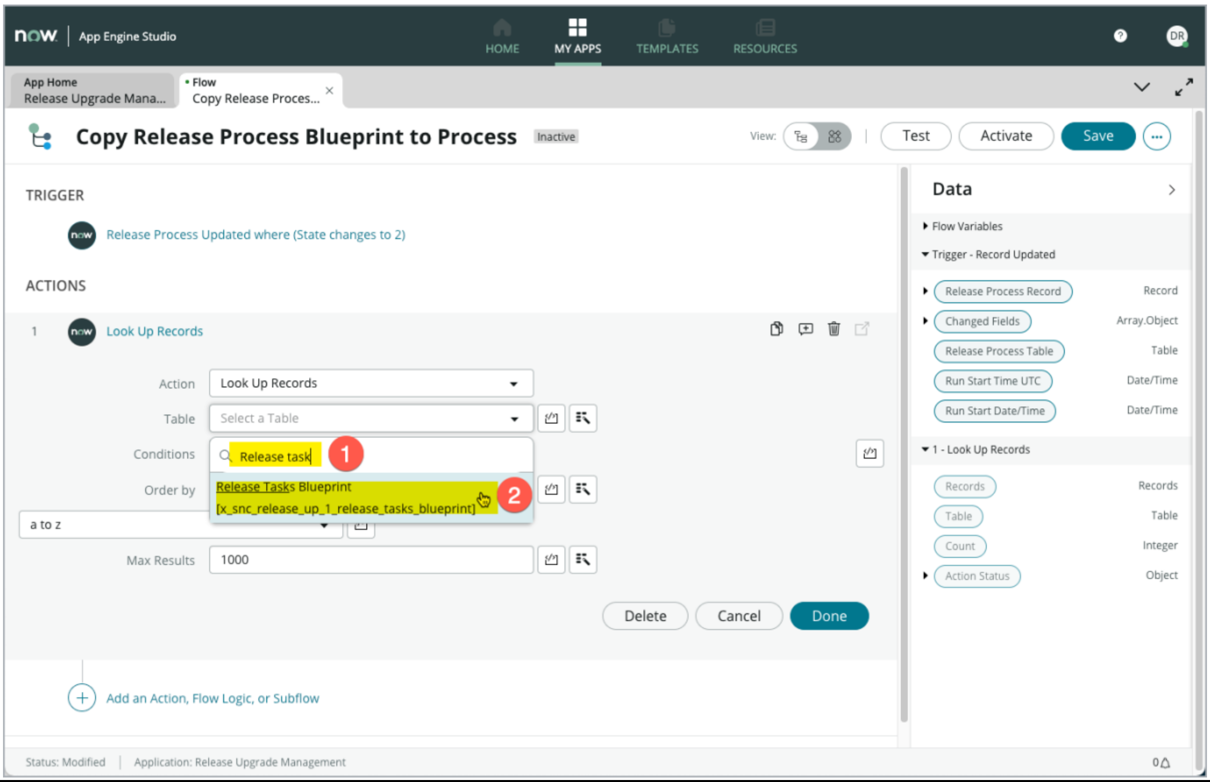


The “Look Up Records” Action is inserted to your Flow.

Click the dropdown of the “Table” field (1).

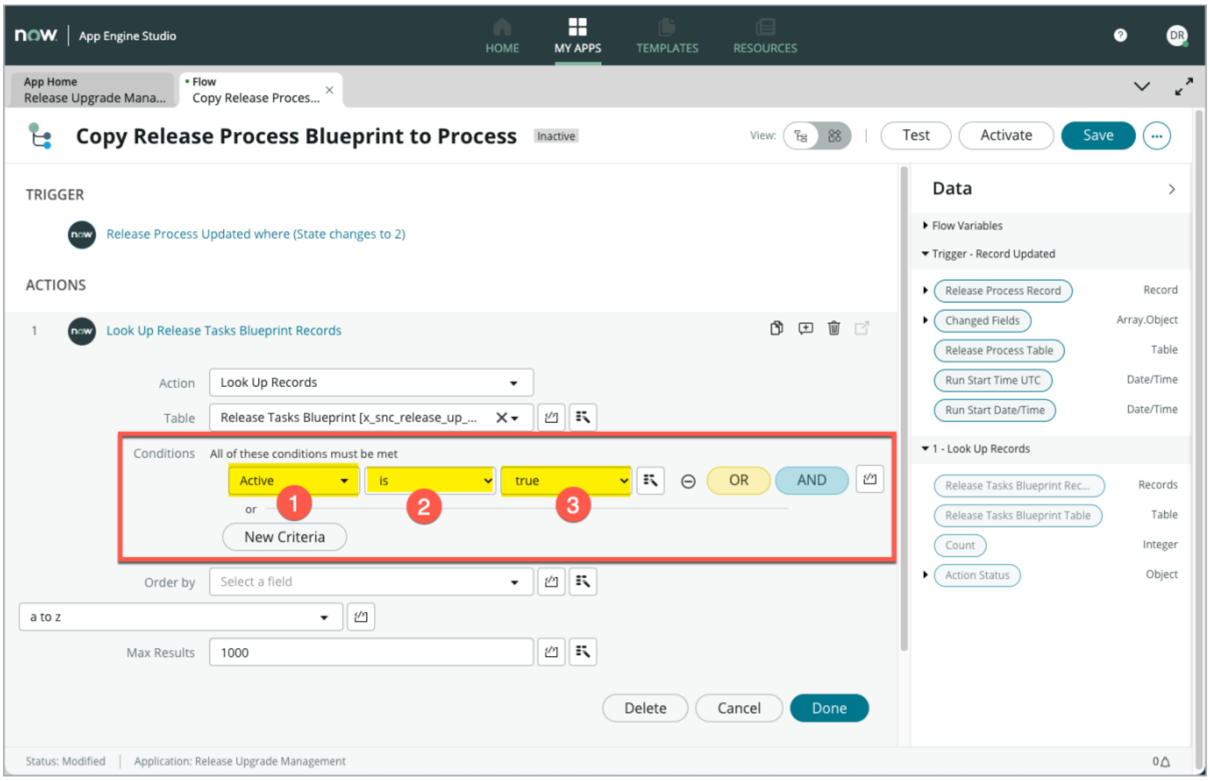


Type “Release task” in the filter (1) and select the “**Release Task Blueprint**” table (2).



The **Condition Builder** is now shown below the Table field.

Select the field “**Active**” (which comes from the “Release Tasks Blueprint” table) in the first dropdown (1), select “**is**” in the second dropdown (2) and “**true**” in the third dropdown (3).

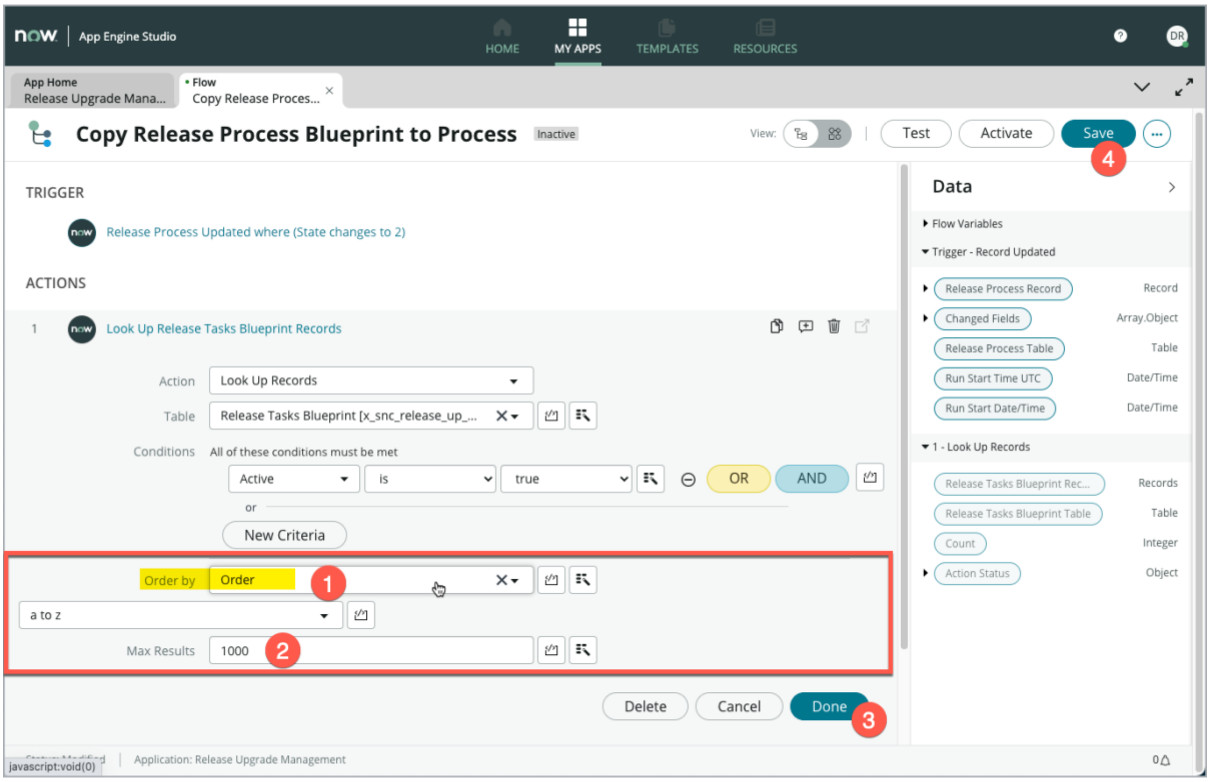


Optionally, select the “Order” field in the “Order by” dropdown (1).
Make sure to leave the “Max Results” field untouched (2).

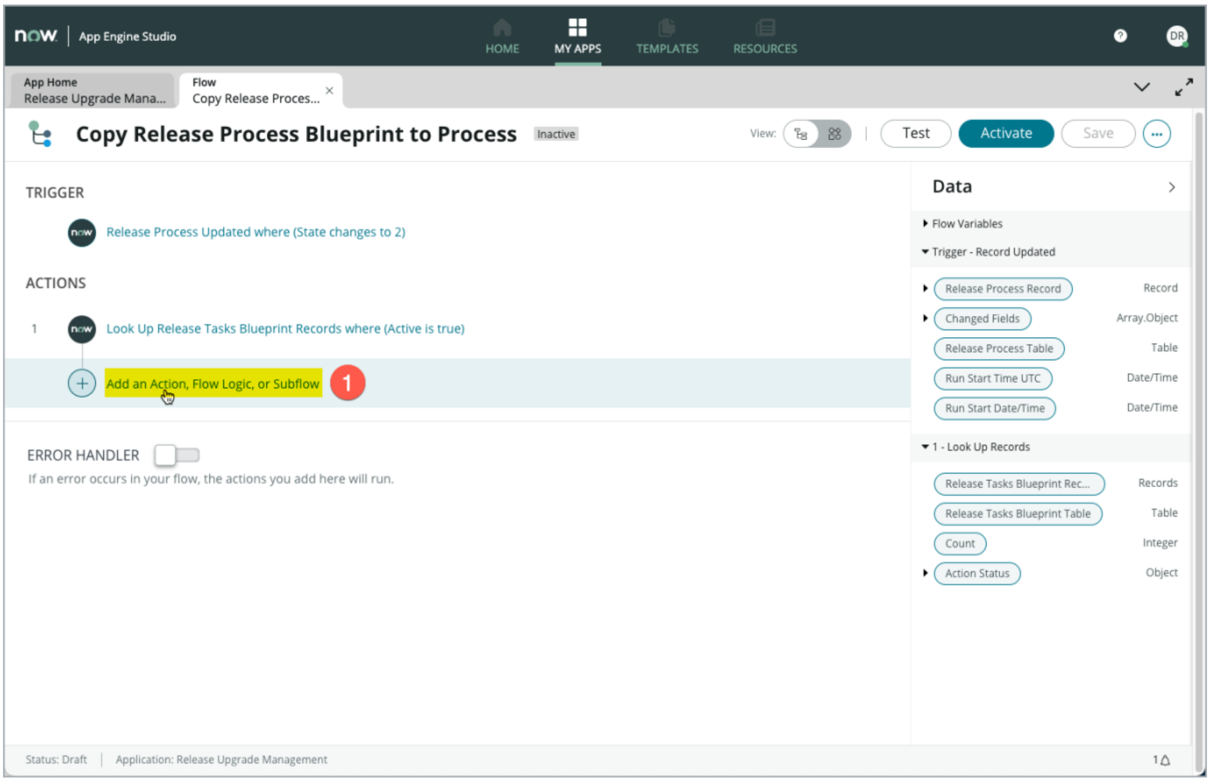
Click “**Done**” (3) when finished.

Click “**Save**” (4) to make your changes to the Flow permanent!

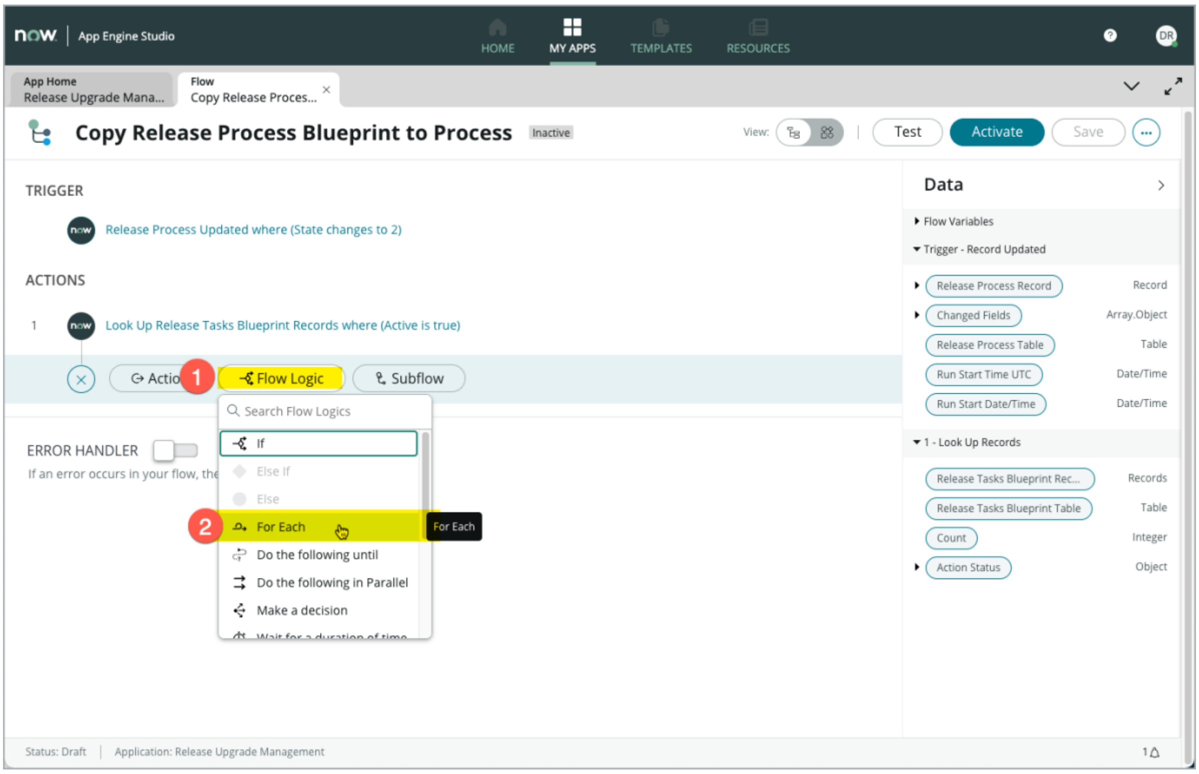
Hint: Save often!



Below your first Action, click "**Add an Action, Flow Logic, or Subflow**" (1)

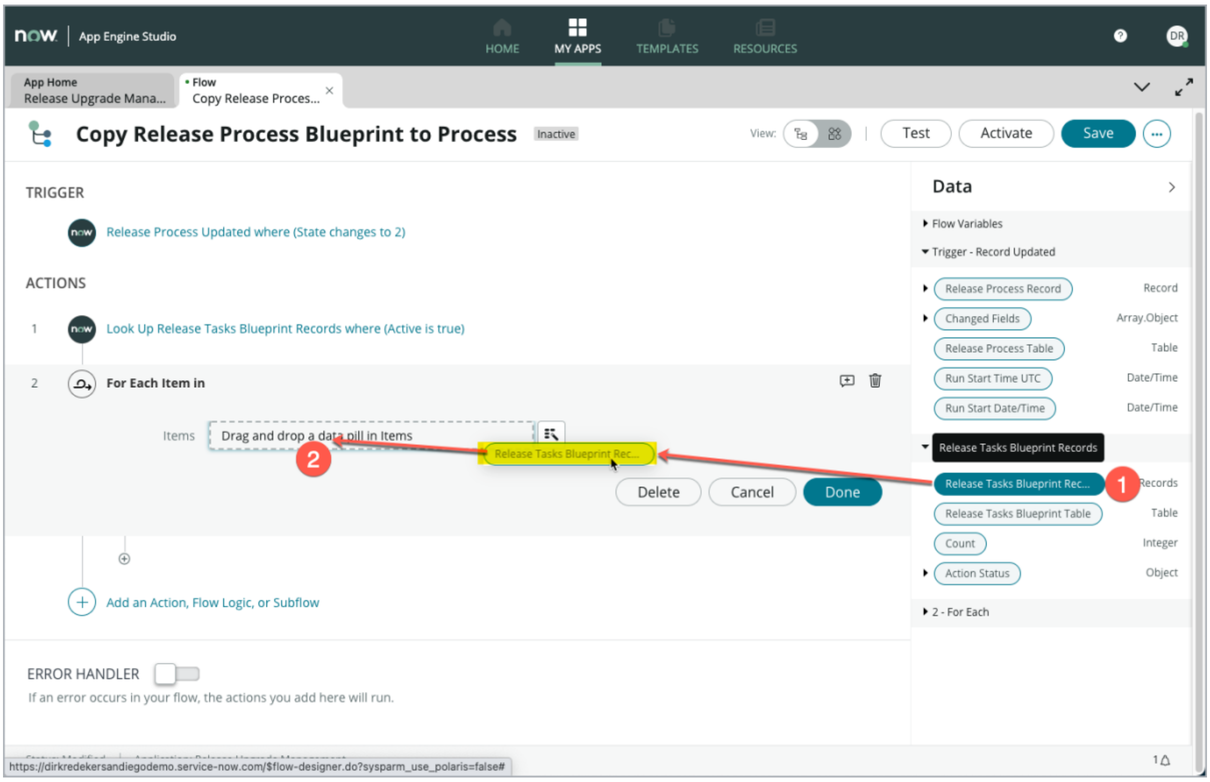


This time, select "**Flow Logic**" (1) and from the dropdown, select "**For Each**" (2).

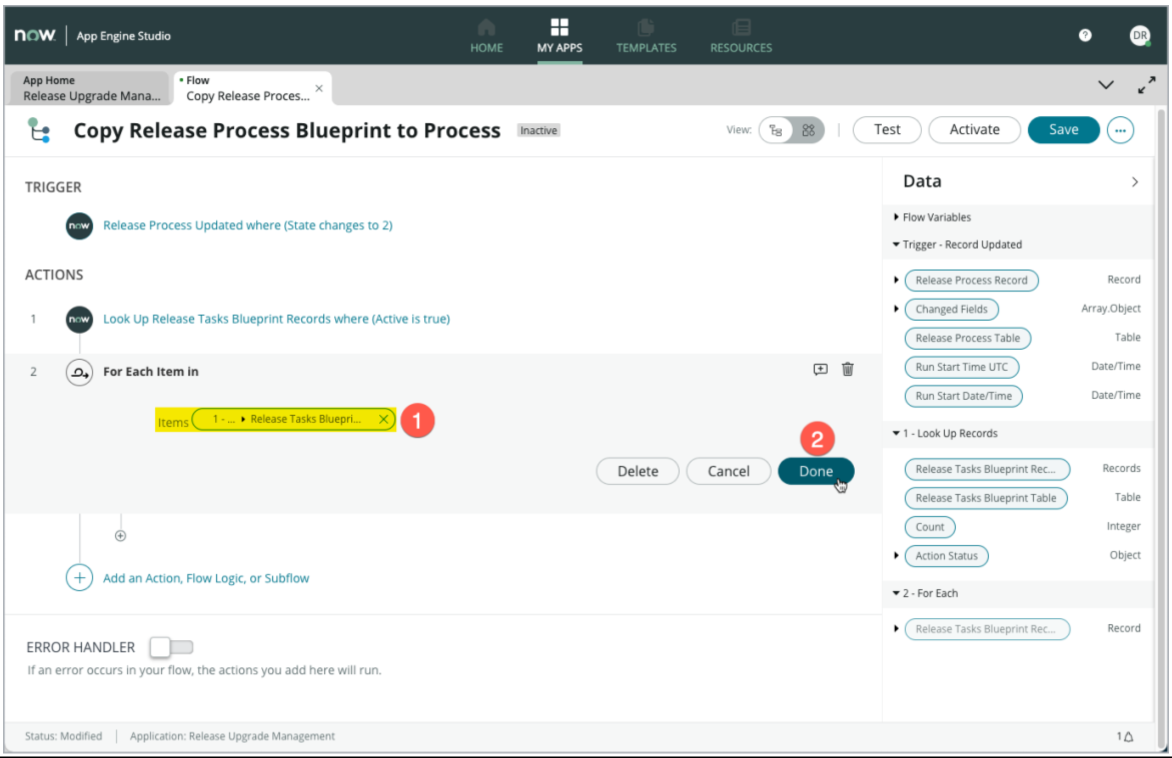


The “For Each” Flow Logic will be added to your flow

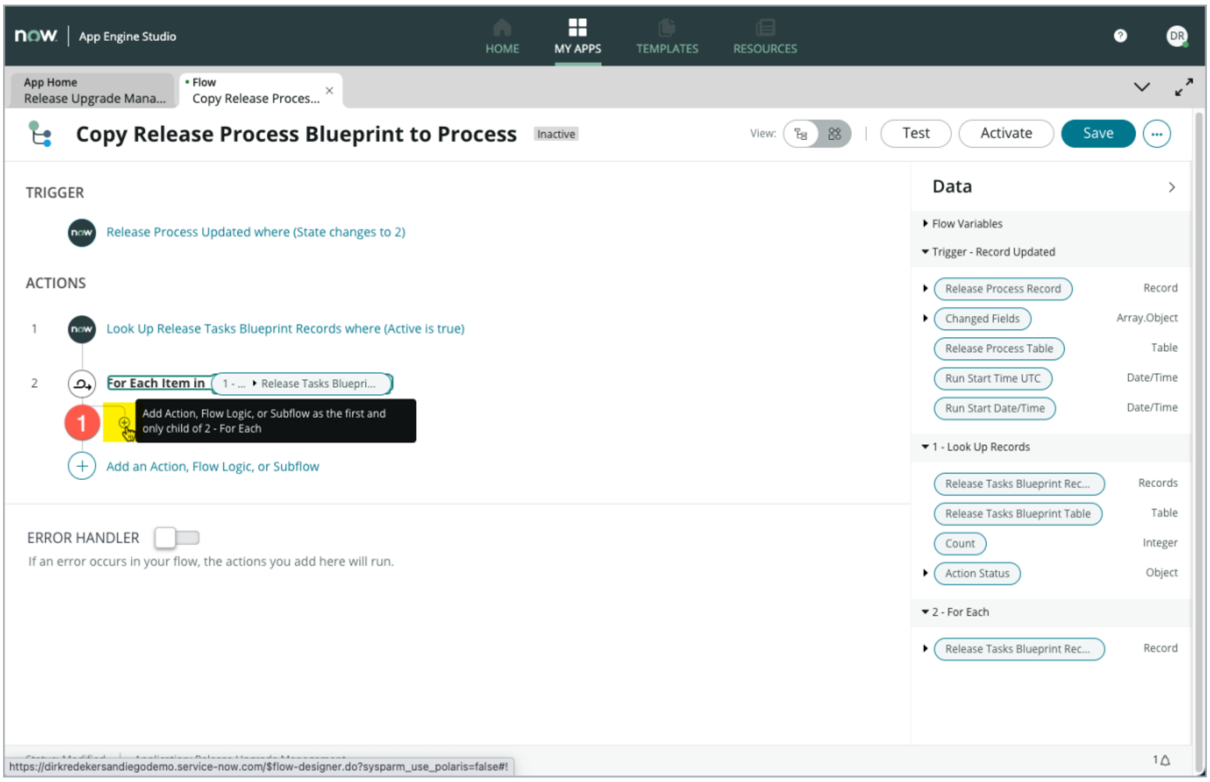
Next, **drag and drop** the “Release Tasks Blueprint Records” data pill (1) from the right-hand data panel to the “**Items**” field of your “For Each” Flow logic (2) – see screenshot below.



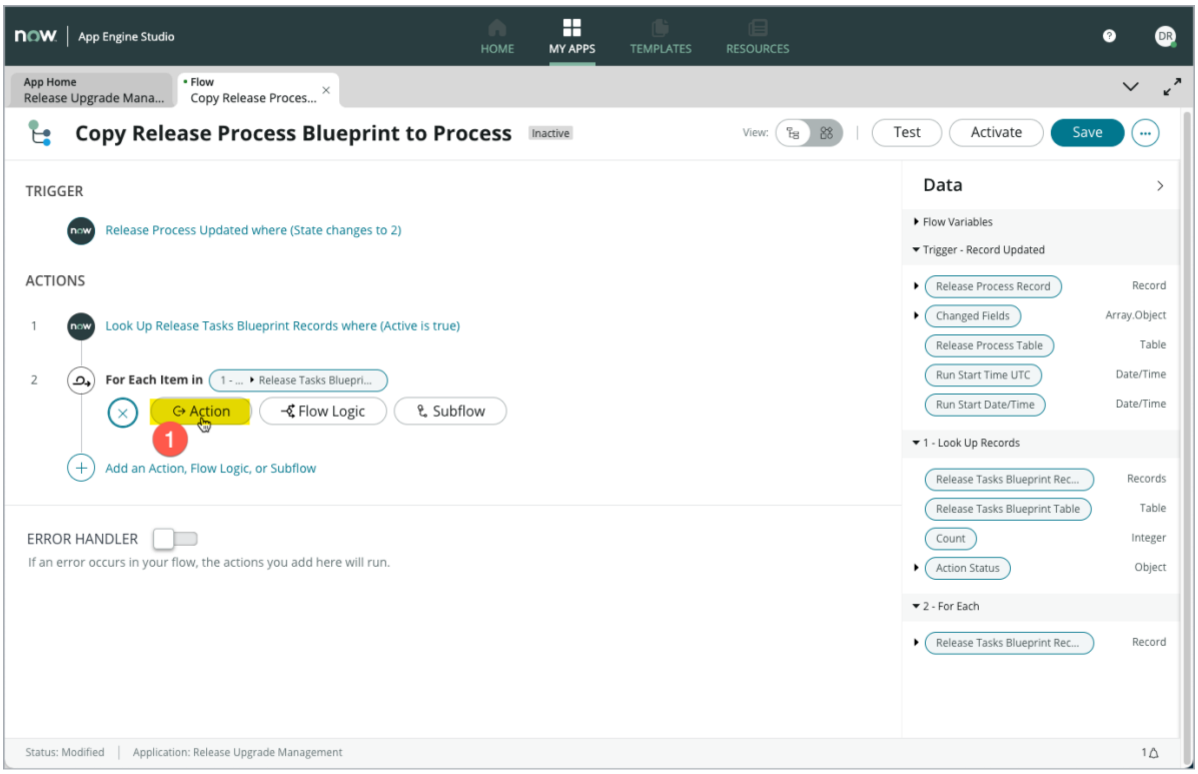
In the next screenshot you can see that the data pill is now used for the Items in the “For Each” Flow logic (1). Click “**Done**” (2) when finished.



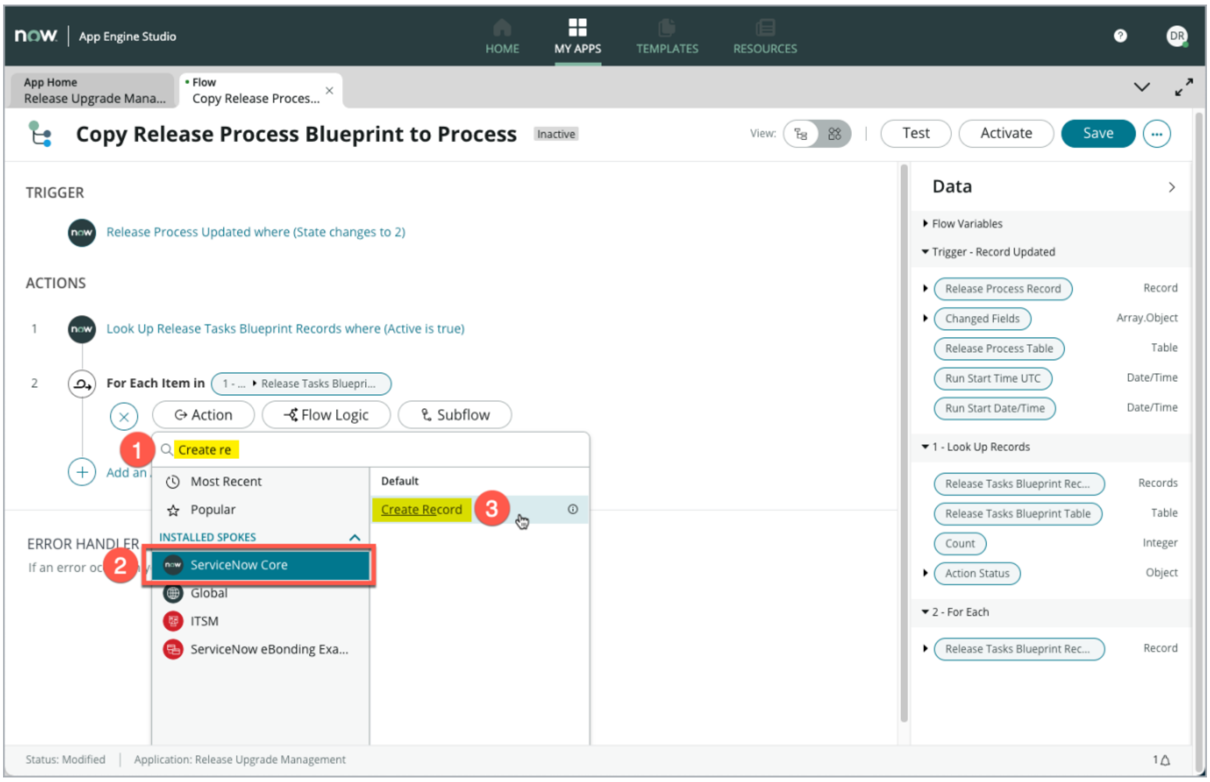
Click the small (+) plus icon (1) below the "For Each Item In" Flow Logic.
This will add an action that will run for EACH item (Record) used in the "For Each" loop.



Select "Action" (1) to insert an Action at this point in the Flow.

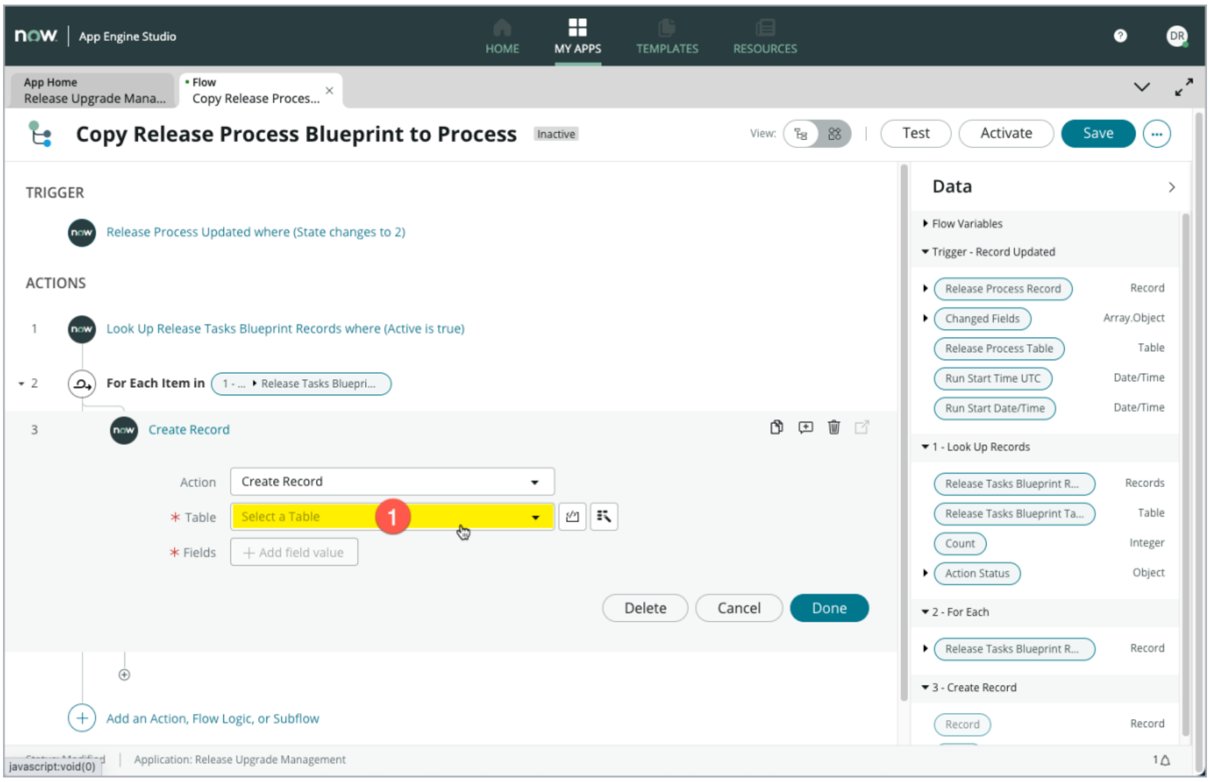


Type “Create re” in the filter (1) and select the “ServiceNow Core” Spoke in the left-side list (2). Find and select the “Create Record” Action in the right-side list (3).

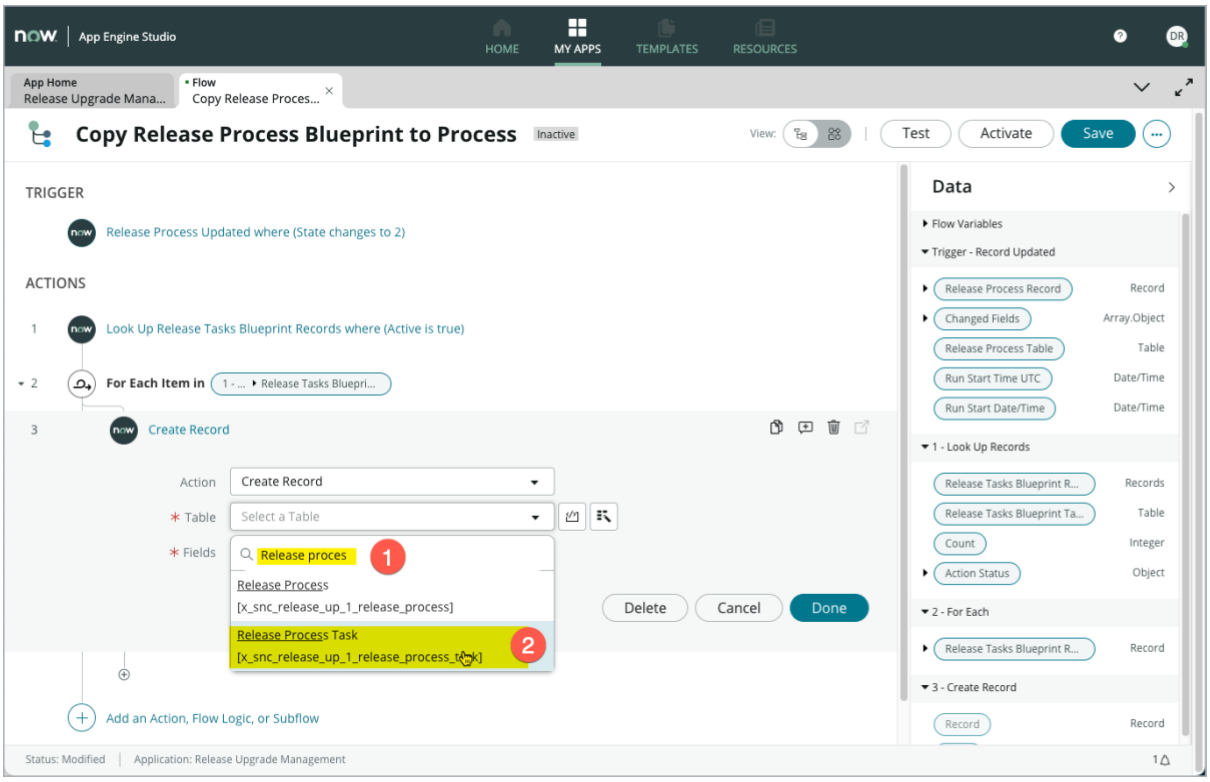


That inserts the “Create Record” Action to your Flow. Note, that this Action is indented below the “For Each” Flow Logic.

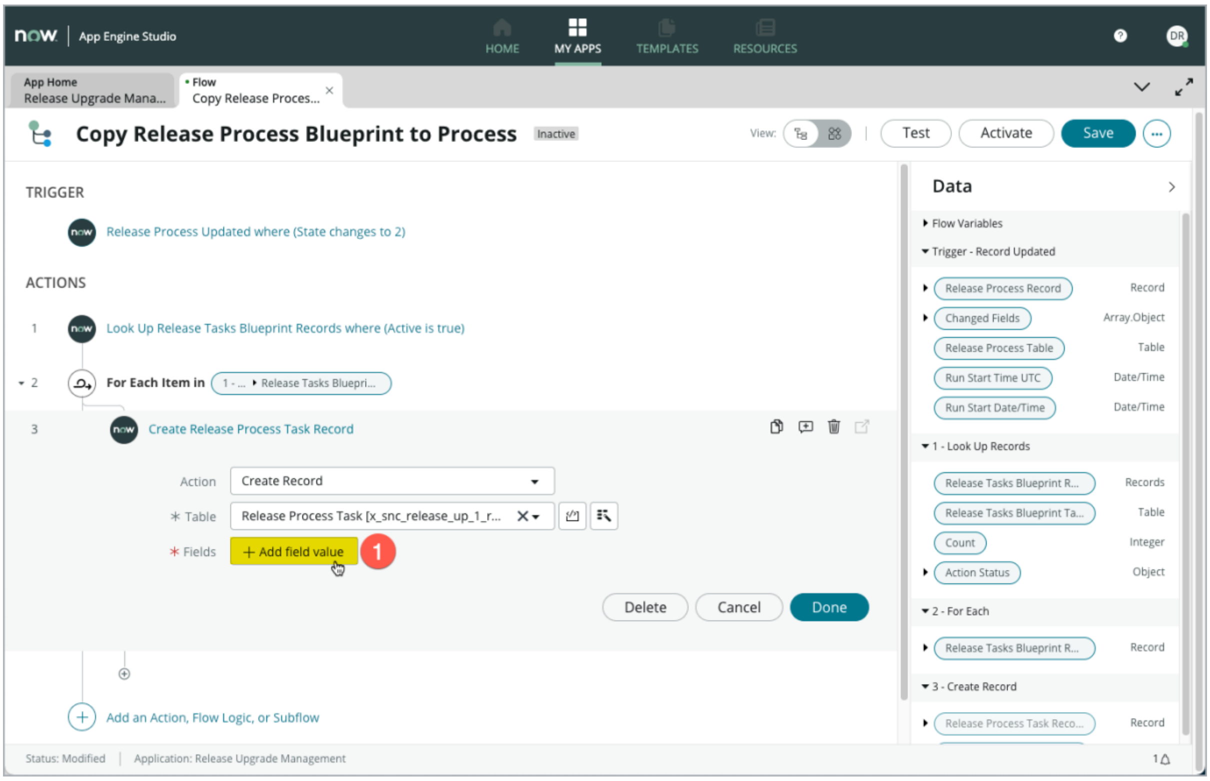
Click the dropdown of the “Table” field (1).



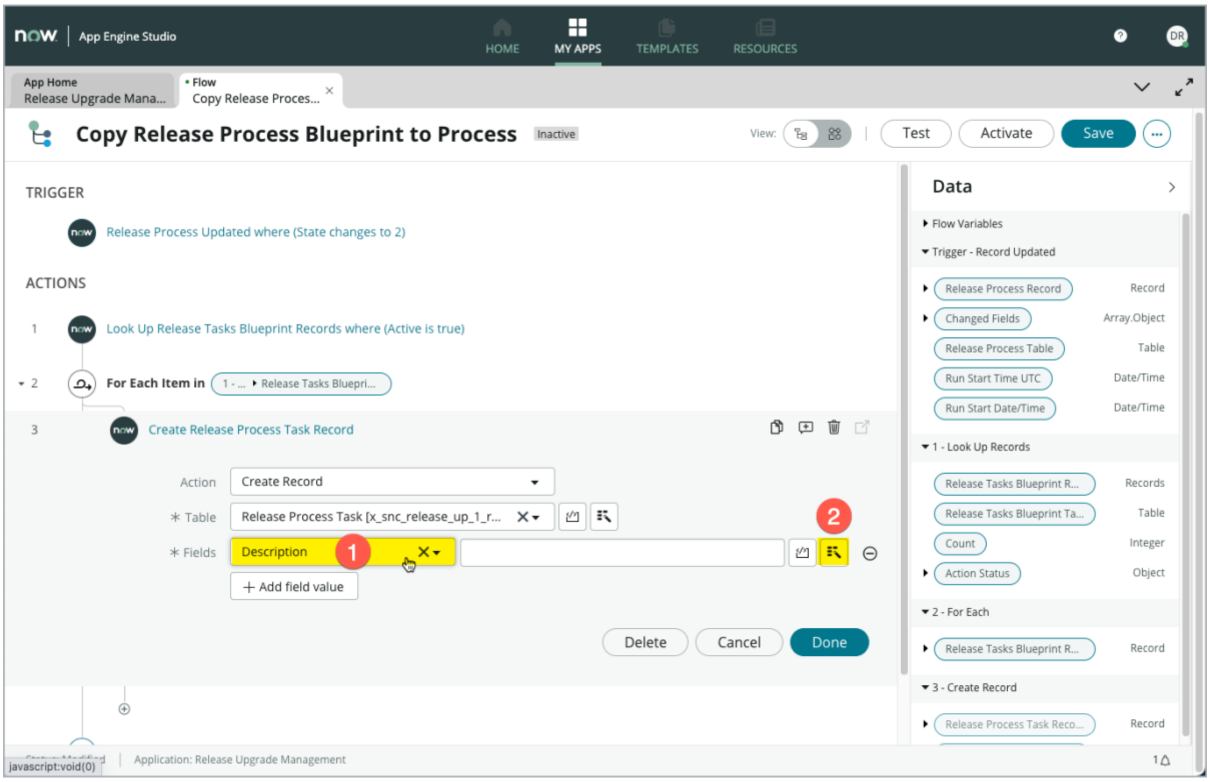
Type “Release process” into the filter (1) and select the “Releaser Process Task” (2) table.



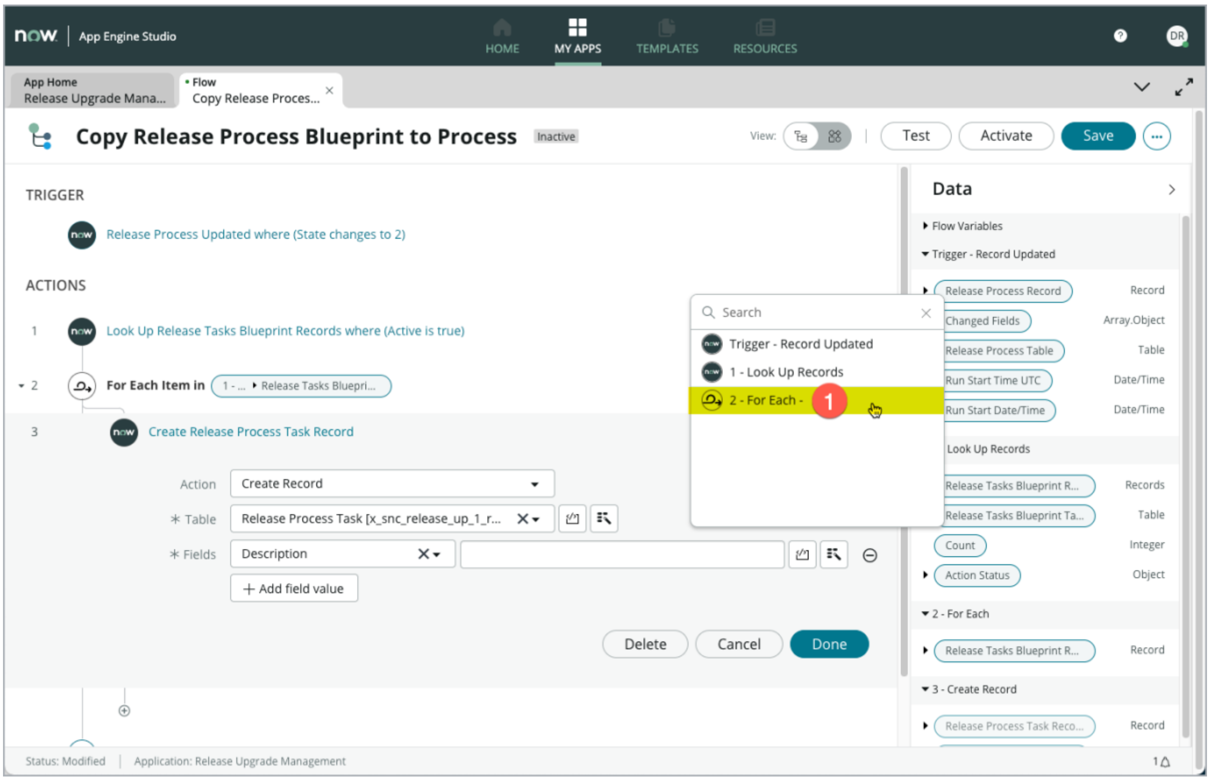
Click “**+ Add field value**” (1)



Select the “Description” field (1) from the List and **click the “data pill picker” symbol** (2).

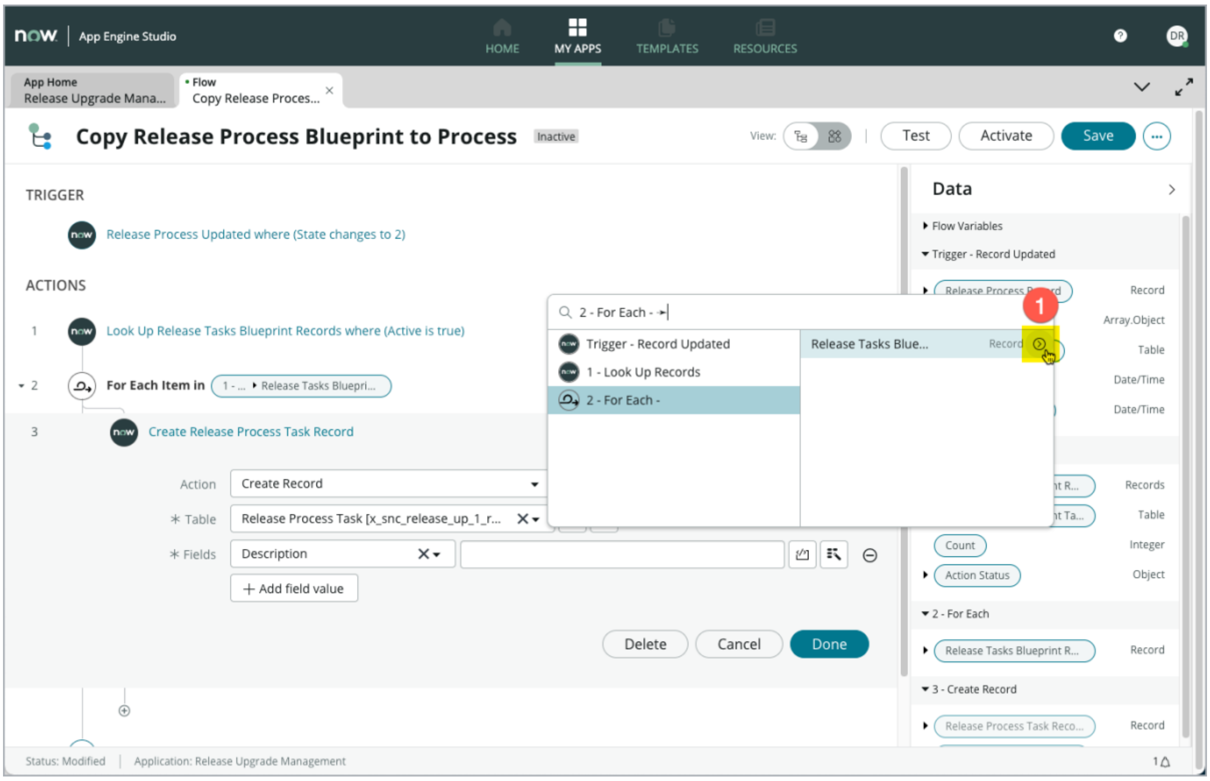


In the Popup Dialog that opens, select the “For Each” entry (1)



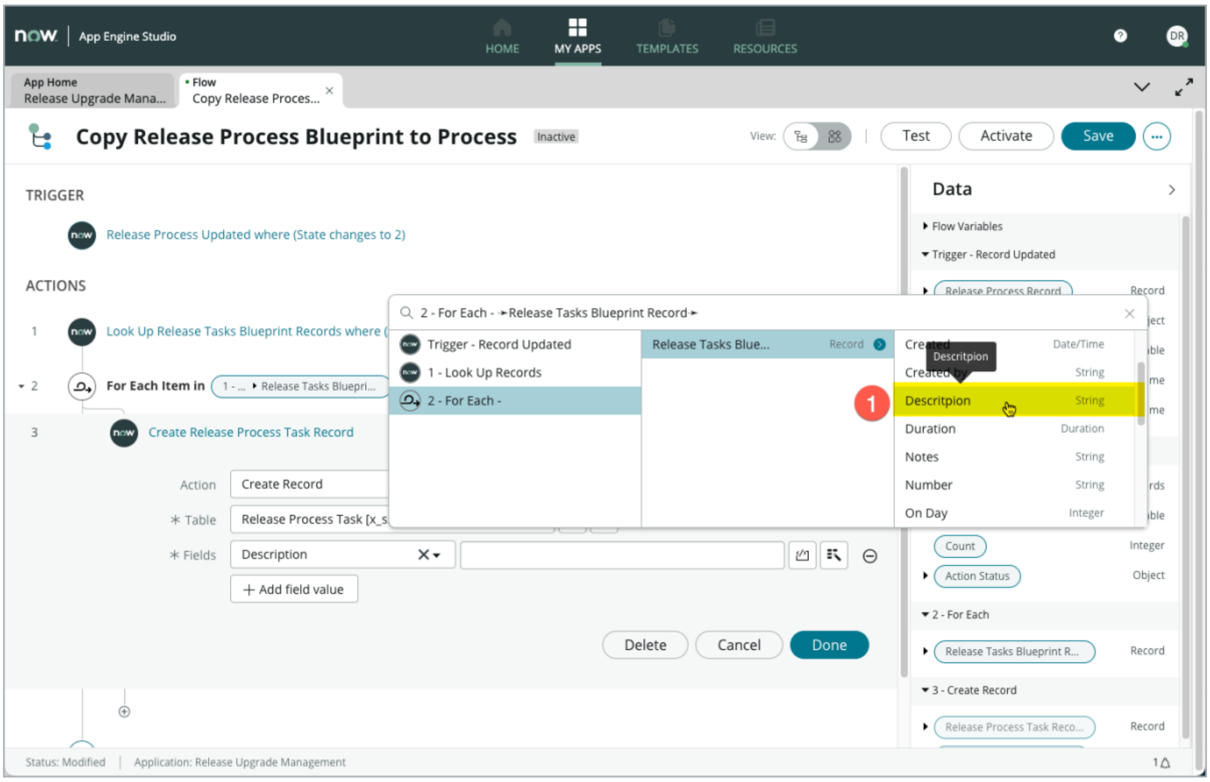
You can see, that the popup dialog expands and shows another list to the right of the selection in the left list. Each list to the right shows' options for the selection of its right-hand list.

In this case, the “For Each” Flow Logic allows to access the “Release Tasks Blueprint” records – one in each loop. Do NOT click the NAME “Release Task Blueprint” in the right-side list. This would SELECT this record. Instead, click the small “>” symbol at the right edge (1).

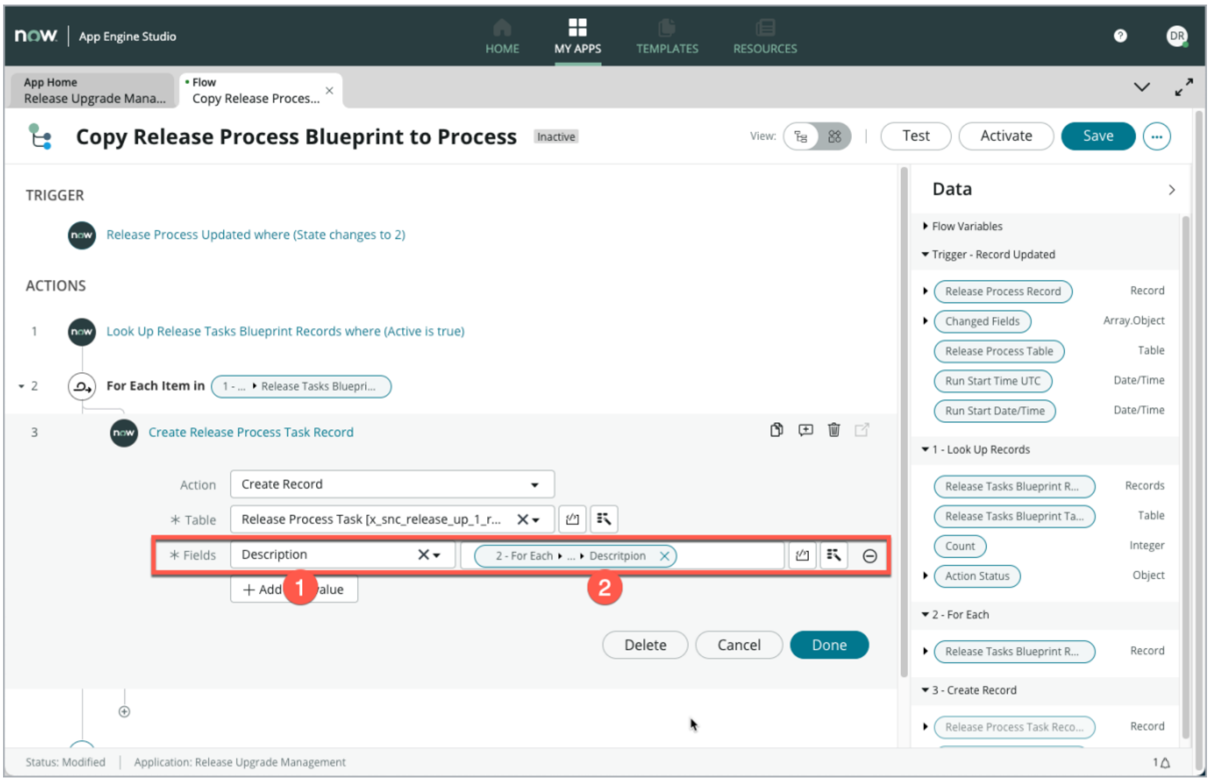


Another list opens to the right in the popup dialog.

From this list, find the “Description” field (1) and select it.



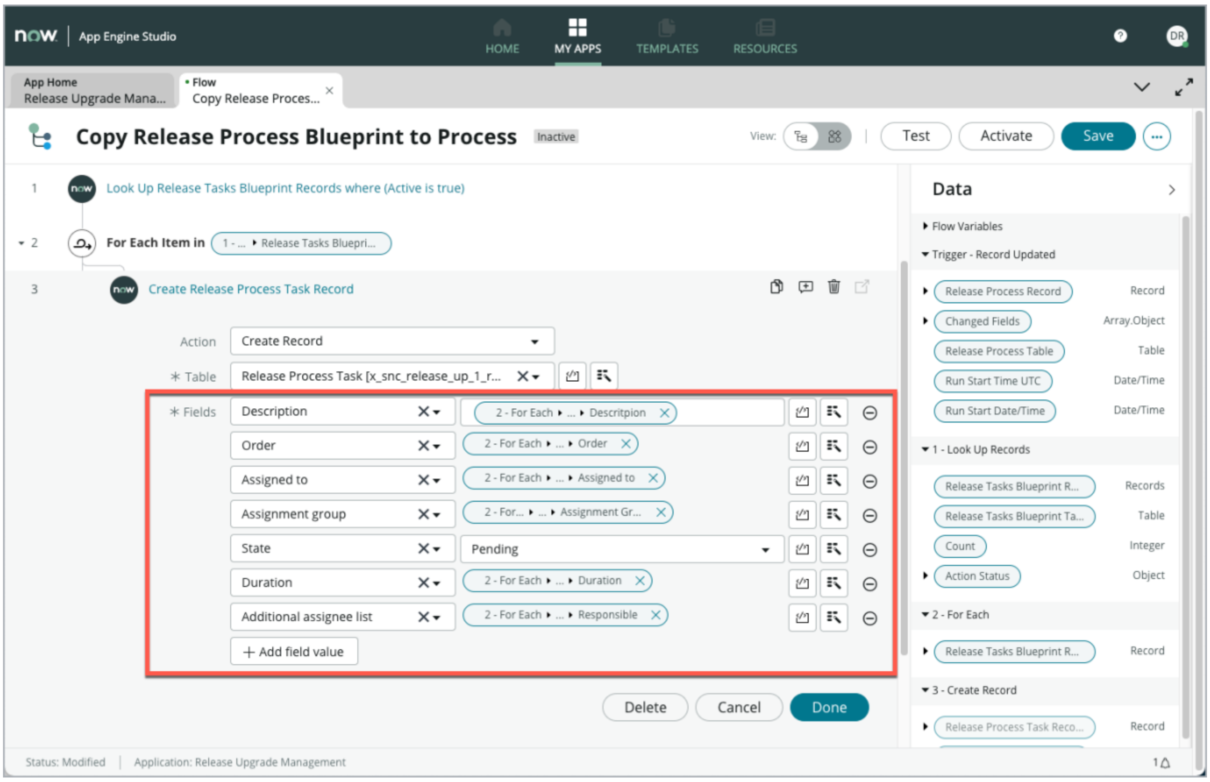
You now have defined, that the Field “Description” (1) in the new record (you are going to create with the “Create Record” Action) is filled with the “Description” (2) field of the “Release Task Blueprint” record in the current Loop.



Now let's **add more fields** for the new record to be created. Click **"+ Add field value**" for each field to fill with the "Create Record" Action.

Look at the screenshot below to review on which and how to fill the fields.

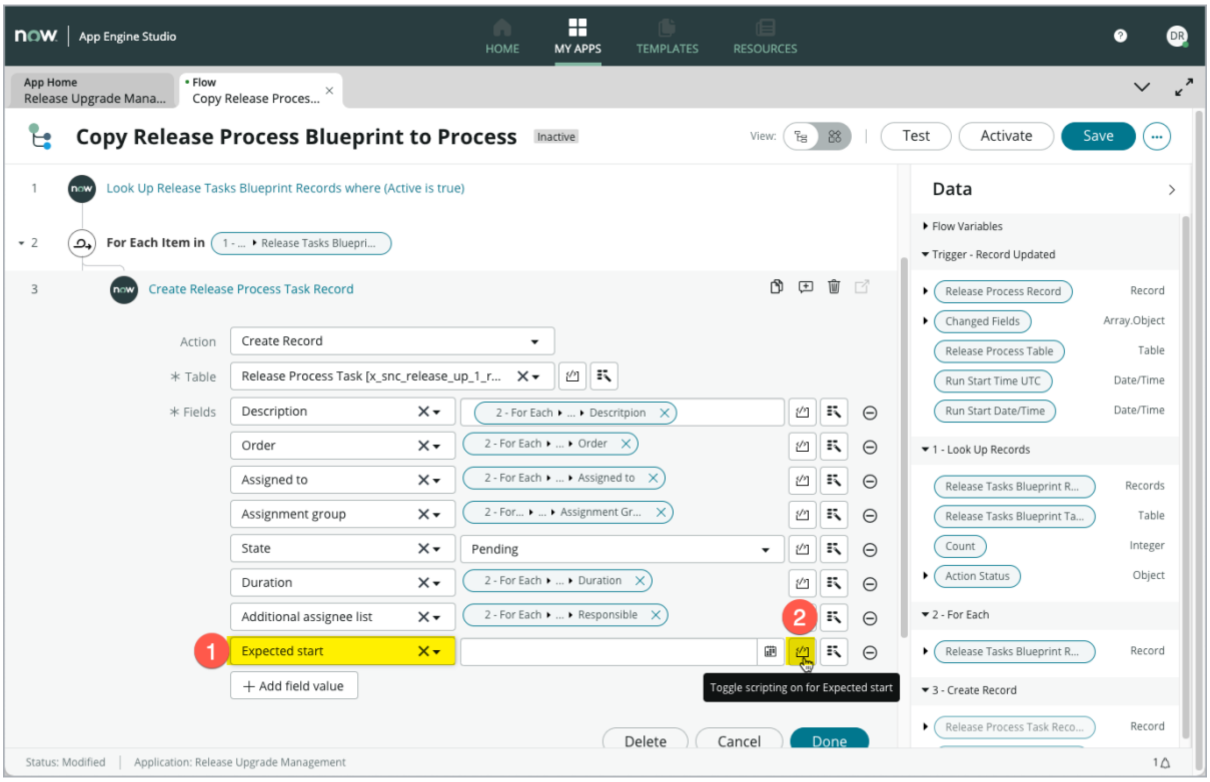
Note:
As there are no step-by-step screenshots for the other field assignments, try to assign them according to the use case. Ask questions in the team, if you need some help.



CHALLENGE

This step is optional: You can also fill in the "**Expected start**" DATE & TIME, which can be calculated, based on the "Expected start date" of the "Release Process" PLUS the number of Days in the "On Day" field in the "Blueprint" table record

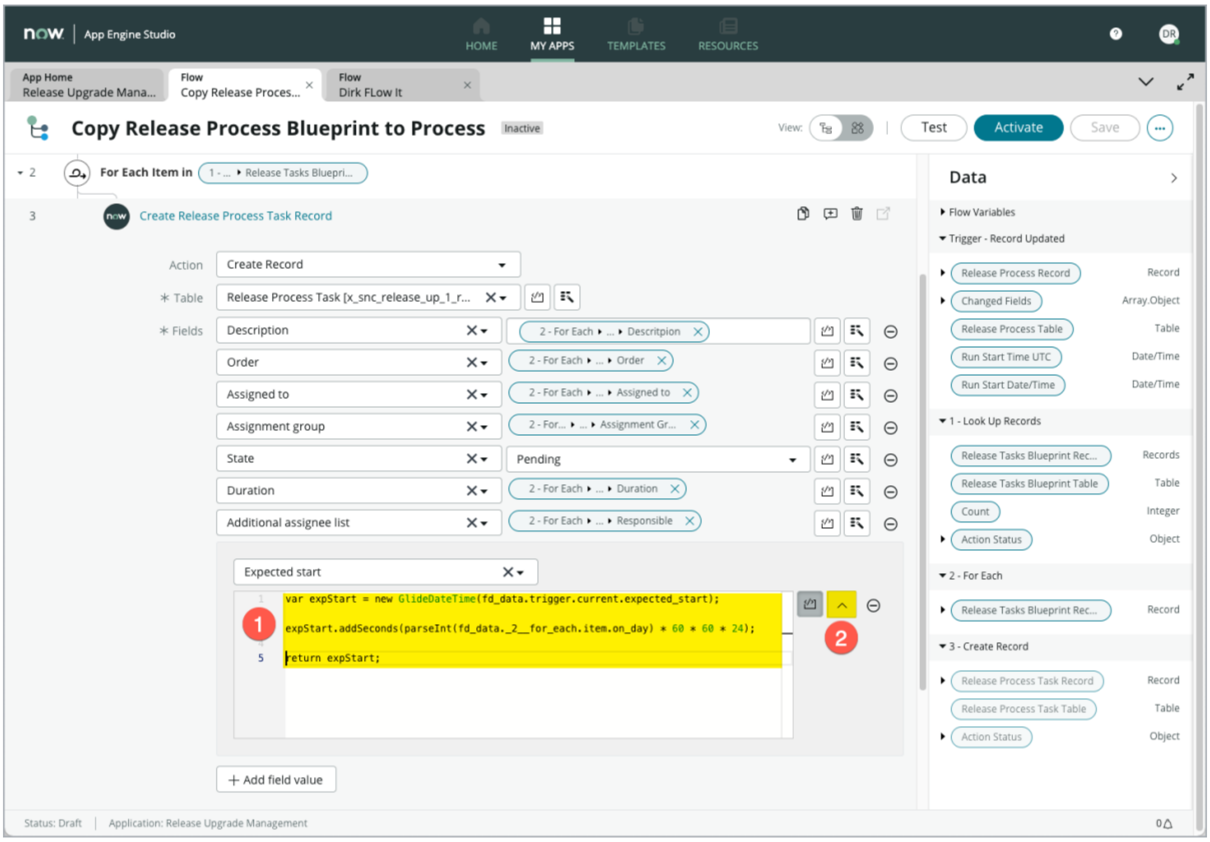
Select the "Expected start" field (1) and **click the "Toggle scripting"** button (2).



Type or copy the following code into the script field (1).

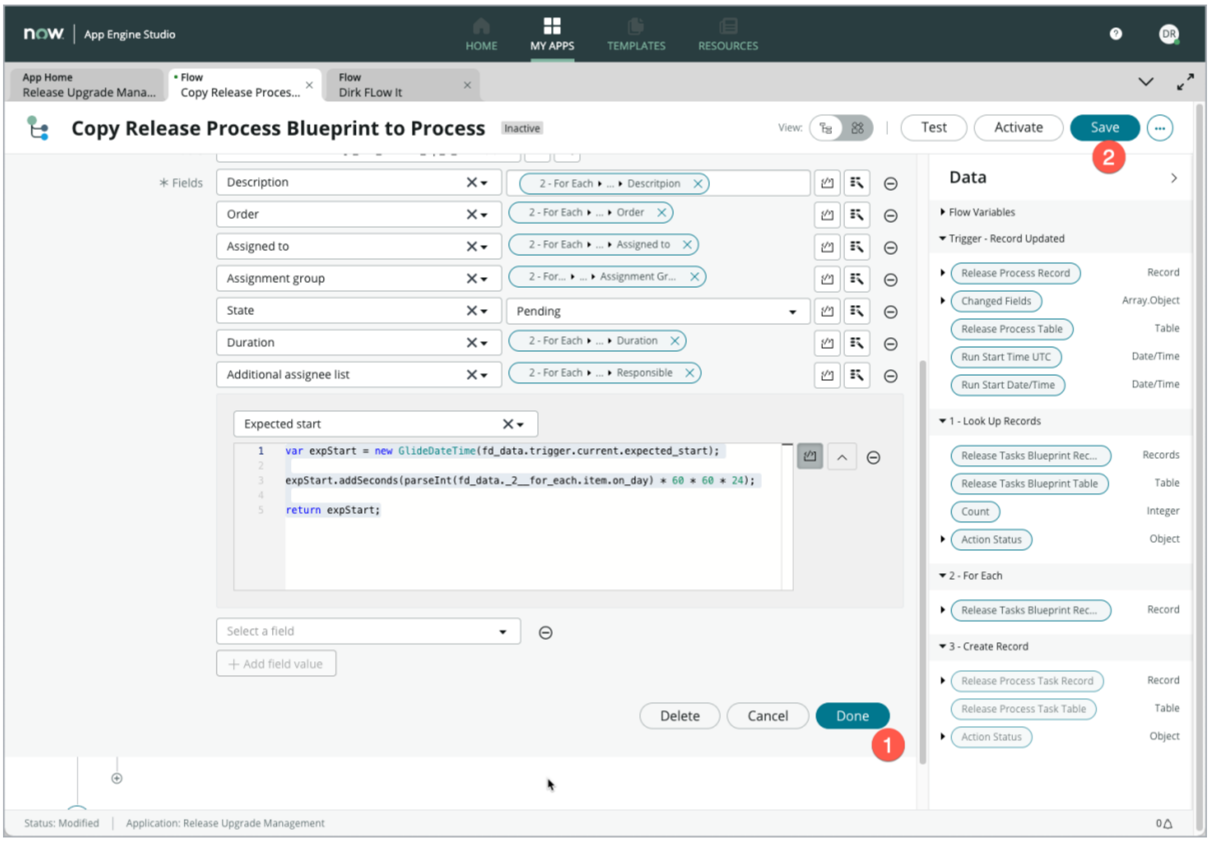
```
var expStart = new GlideDateTime(fd_data.trigger.current.expected_start);

expStart.addSeconds(parseInt(fd_data._2__for_each.item.on_day) * 60 * 60 * 24);

return expStart;
```

You can open/close the script field using the button indicated in the screenshot below (2).



Click “Done” (1) when finished.

Save your changes clicking “Save” (2) in the upper-right corner

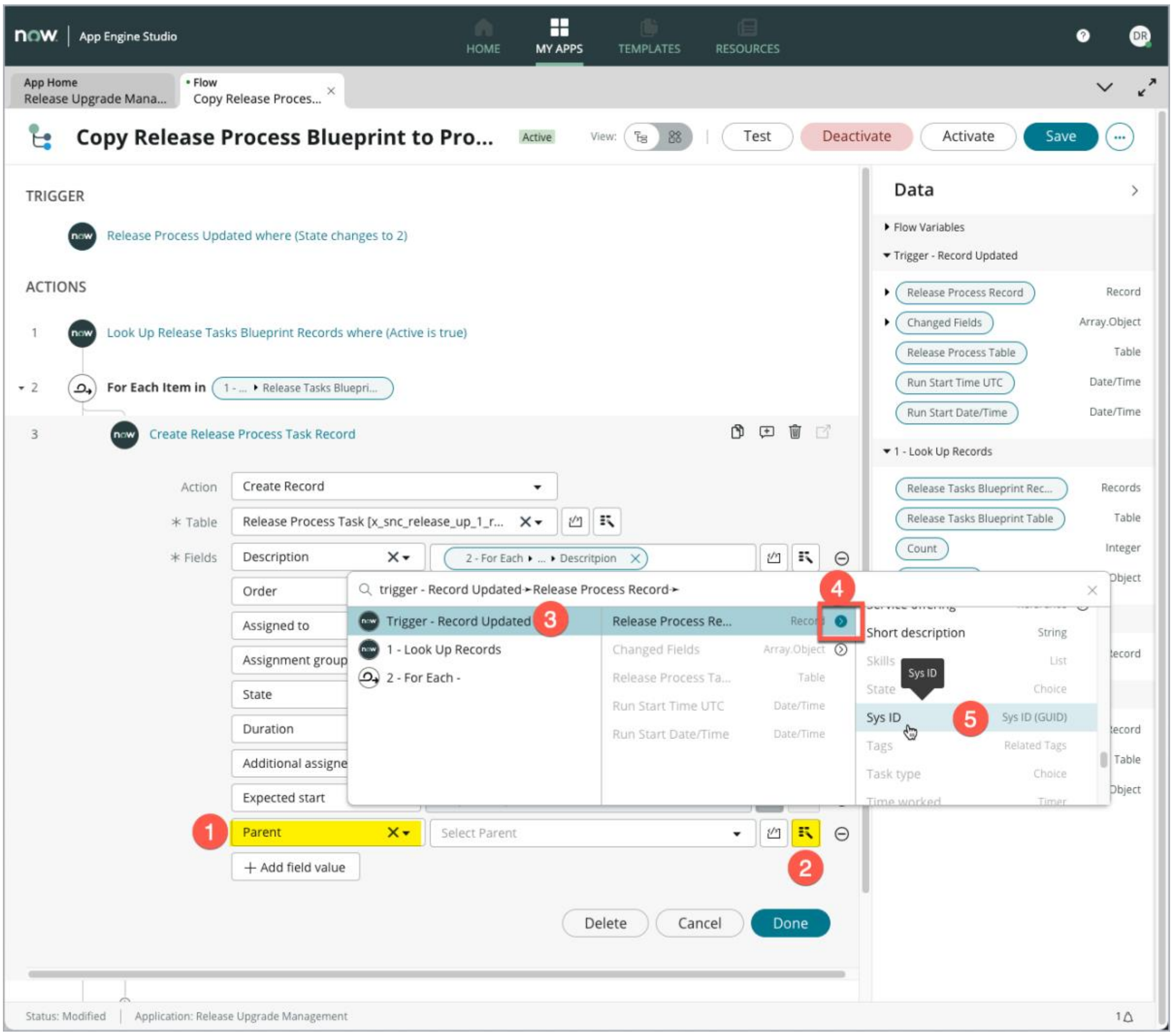


END OF OPTIONAL STEP

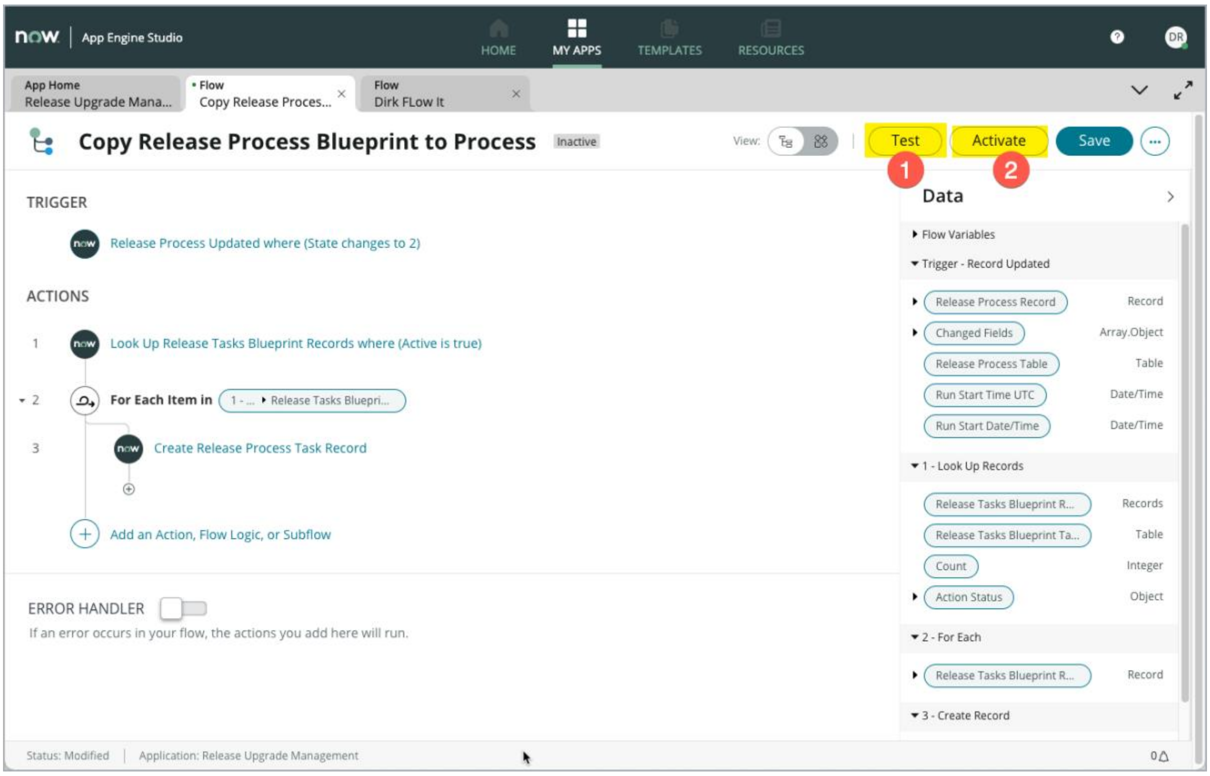
NOTE:

This assignment is essential to link the “Release Process Task” Records to the main “Release Process” Record (as “child” records – so that the will appear in the Related List)

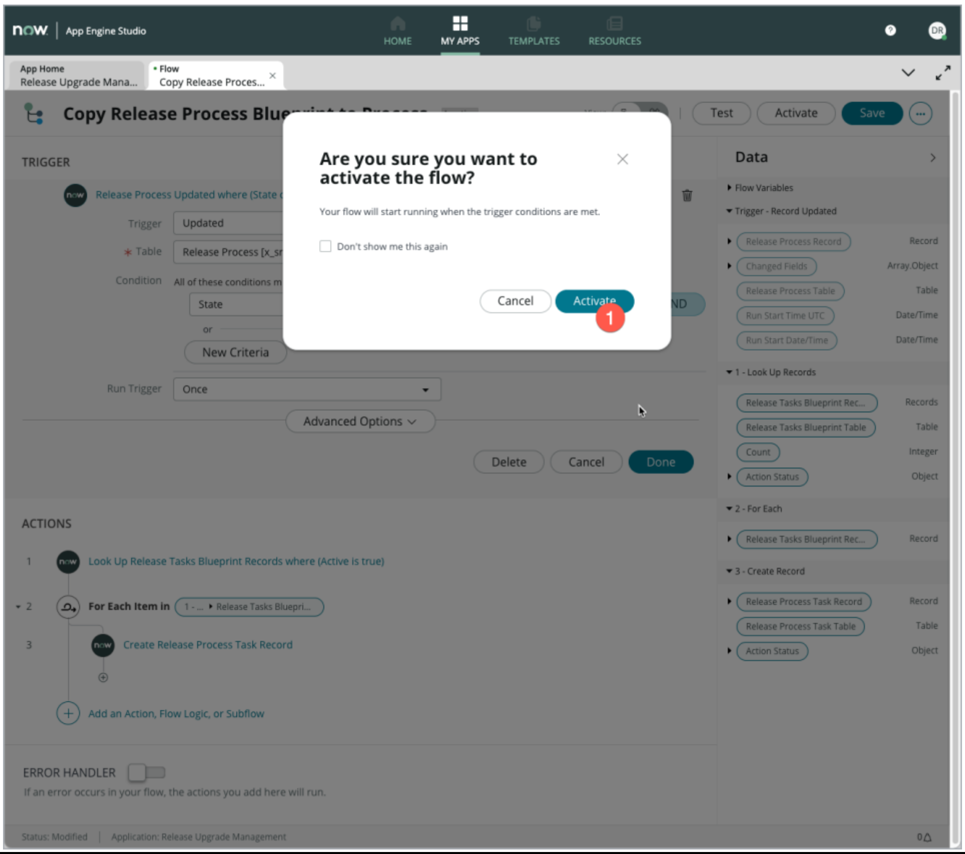
The next step links the Task Record to the main Release Process using the “Parent” field.



You can now Test (1) your Flow or Activate (2) it.
For now, we just click "Activate" (2) to allow triggering the Flow in your App at runtime.

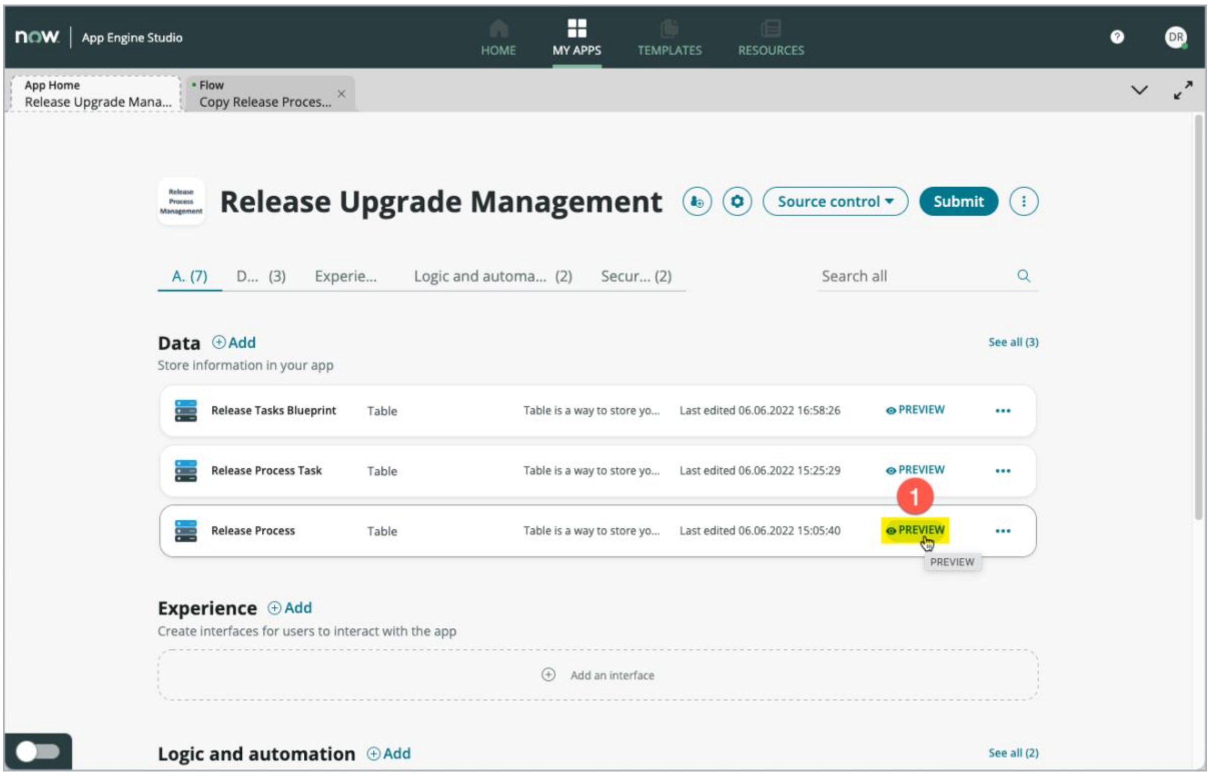


Click "**Activate**" (1) to confirm your selection

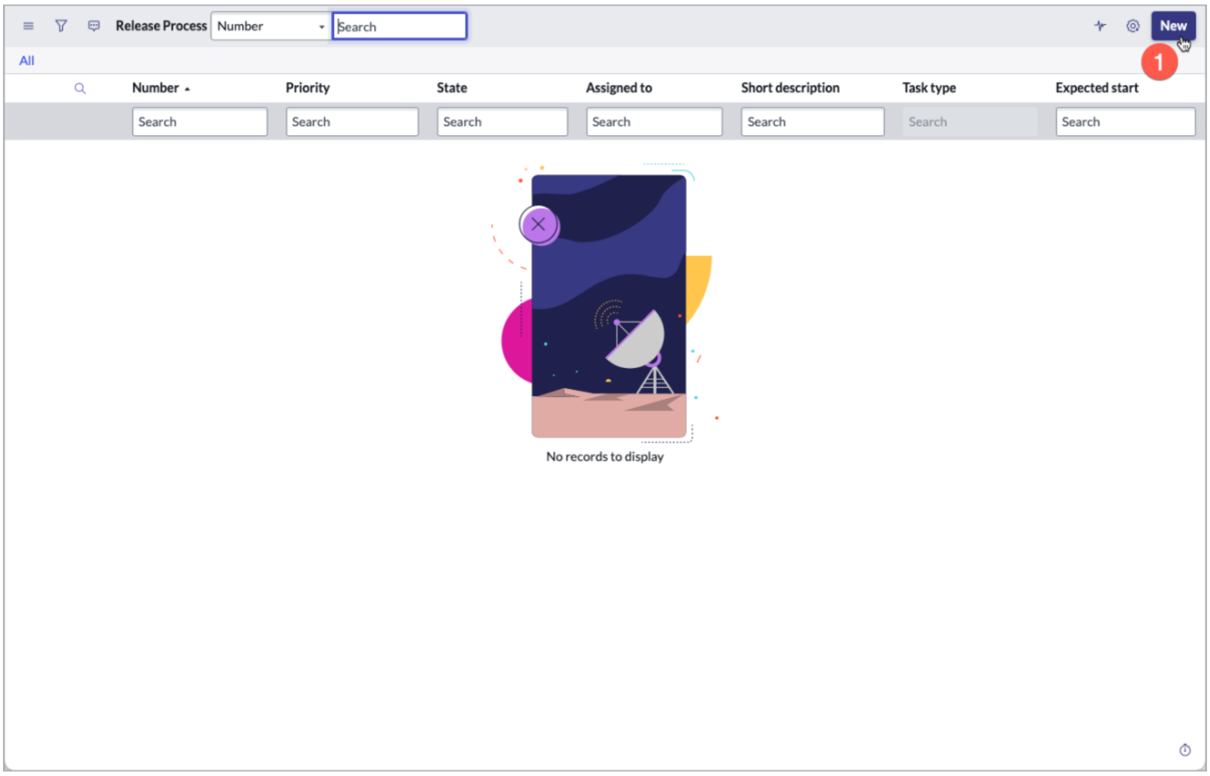


Test the new Flow in the App

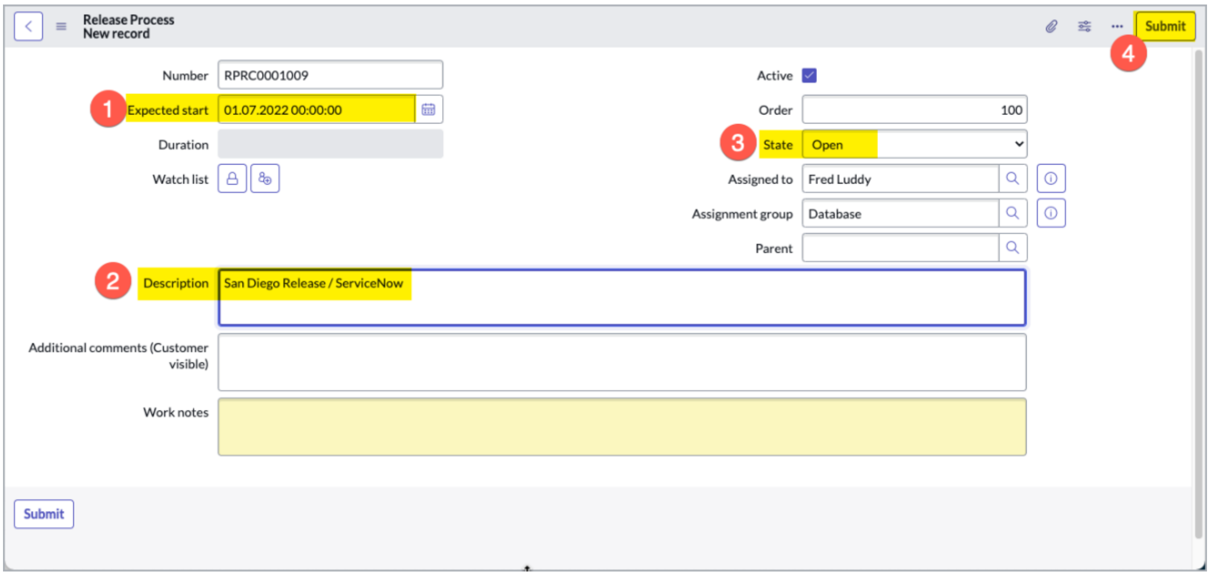
From the App dashboard click “**PREVIEW**” (1) on the “**Release Process**” table (in the “Data” section).



Click “New” (1) in the upper-right corner.



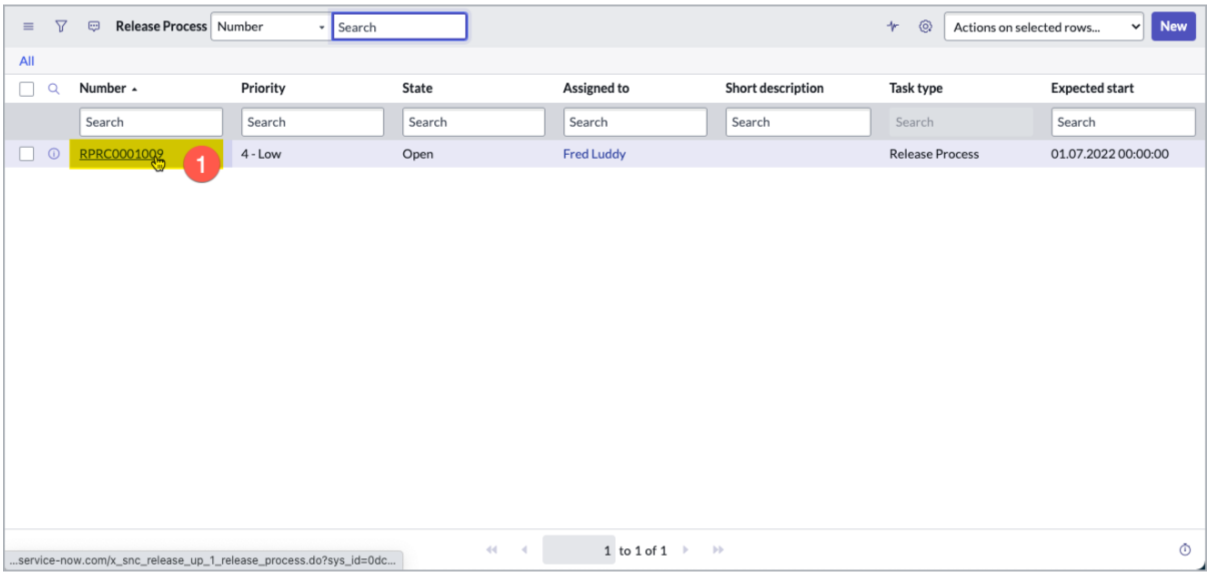
Fill at least the fields indicated in the screenshot below.
Click "**Submit**" (4) to create the record.



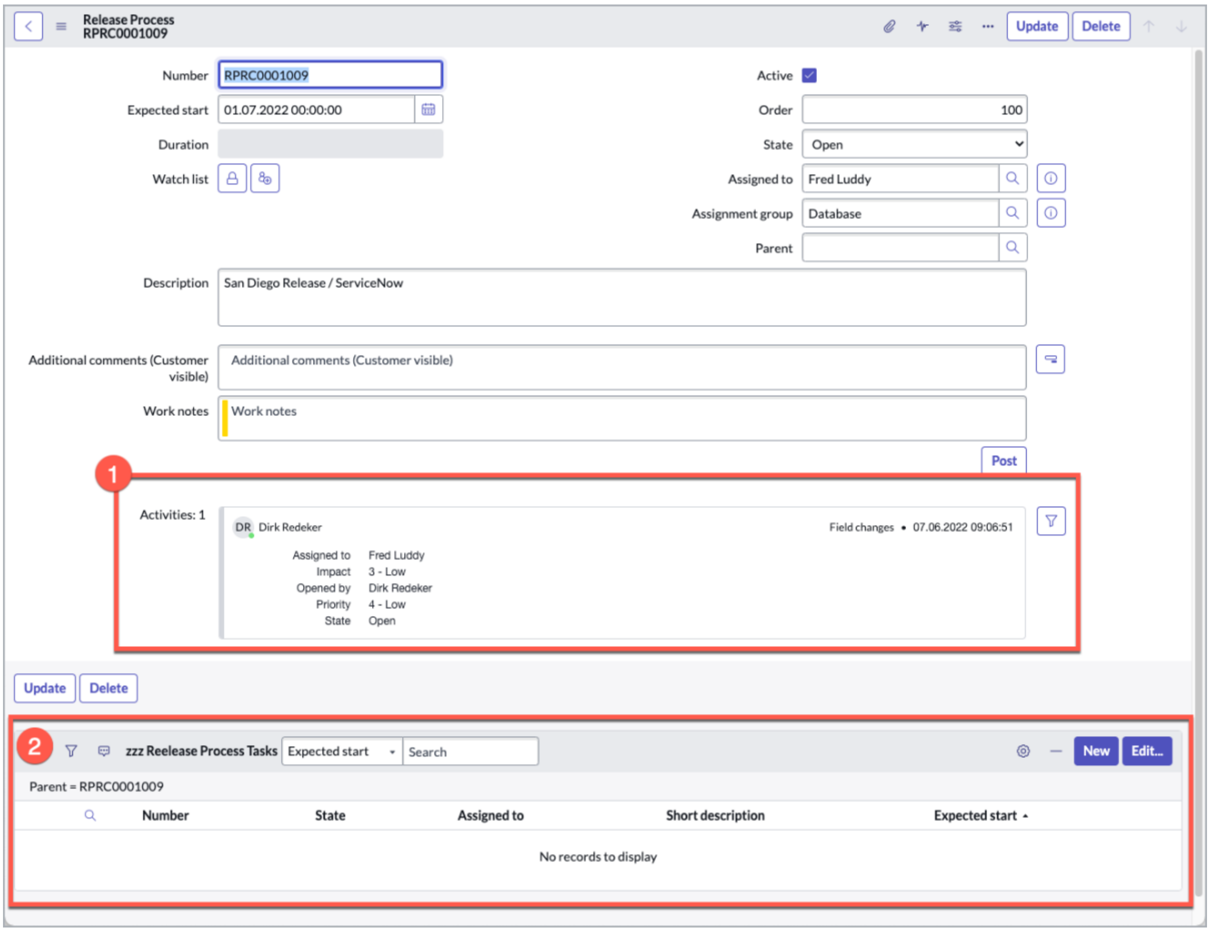
You will be **navigated back to** the List of "**Release Processes**".

Note: The Number field of your Record may be different to the one shown in the screenshot below.

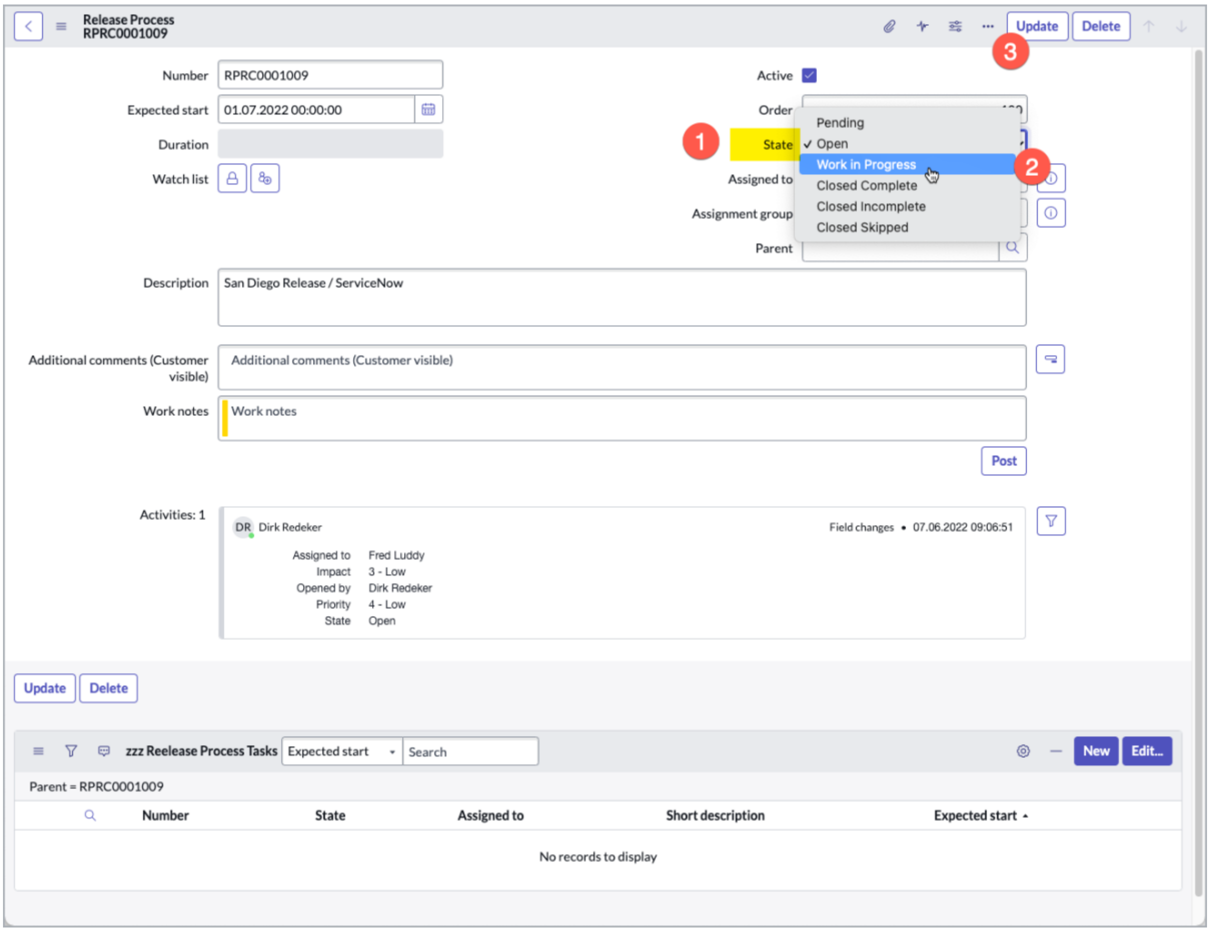
Click the record in the **first column** (1) to open the Record in the Form view.



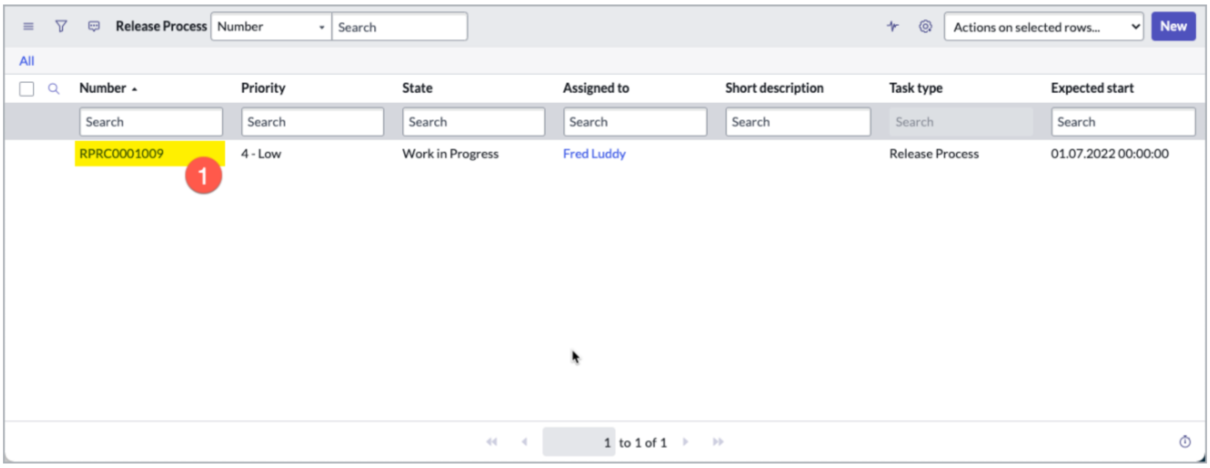
Review the newly created Record.
Note, the Activities automatically logged for your Release Process.
Note, that the Related List is still empty.



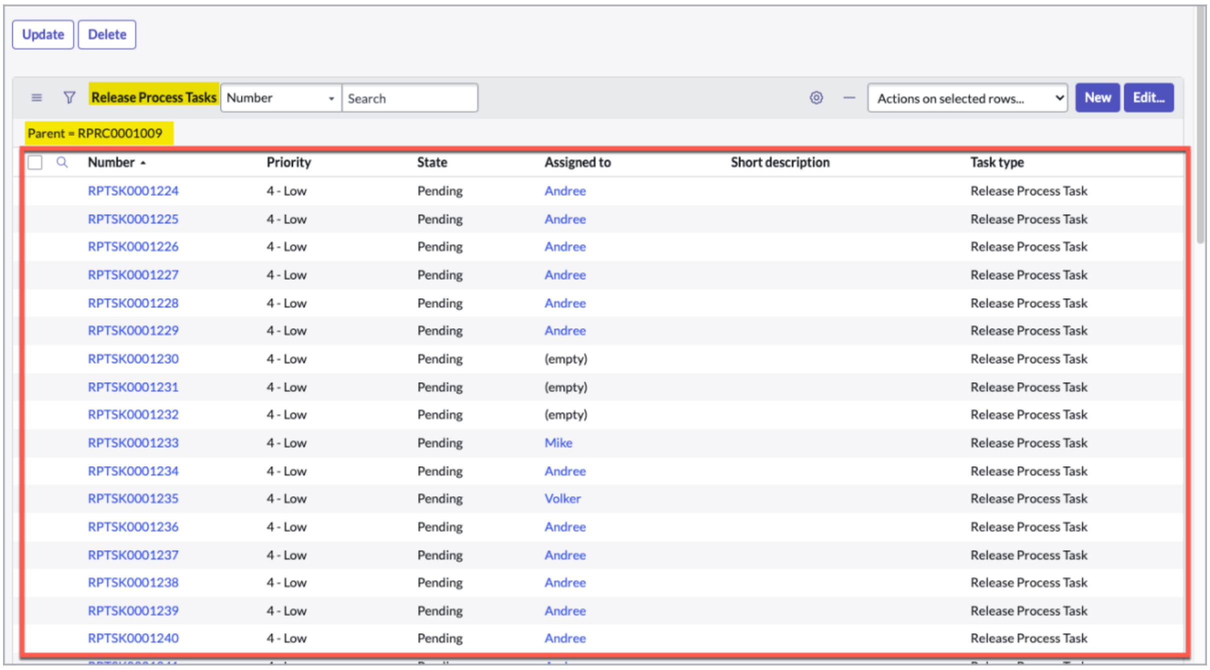
Change the value of the “State” field (1) to “Work in Progress” (2) and click “Update” (3) the Record.
This will start/trigger the Flow we created (State changes to “Work in progress”).



Re-Open the Record from the list (1)



The Related List now contains the new Tasks copied from the Blueprint – see screenshot below.



Number	Priority	State	Assigned to	Short description	Task type
RPTSK0001224	4 - Low	Pending	Andree		Release Process Task
RPTSK0001225	4 - Low	Pending	Andree		Release Process Task
RPTSK0001226	4 - Low	Pending	Andree		Release Process Task
RPTSK0001227	4 - Low	Pending	Andree		Release Process Task
RPTSK0001228	4 - Low	Pending	Andree		Release Process Task
RPTSK0001229	4 - Low	Pending	Andree		Release Process Task
RPTSK0001230	4 - Low	Pending	(empty)		Release Process Task
RPTSK0001231	4 - Low	Pending	(empty)		Release Process Task
RPTSK0001232	4 - Low	Pending	(empty)		Release Process Task
RPTSK0001233	4 - Low	Pending	Mike		Release Process Task
RPTSK0001234	4 - Low	Pending	Andree		Release Process Task
RPTSK0001235	4 - Low	Pending	Volker		Release Process Task
RPTSK0001236	4 - Low	Pending	Andree		Release Process Task
RPTSK0001237	4 - Low	Pending	Andree		Release Process Task
RPTSK0001238	4 - Low	Pending	Andree		Release Process Task
RPTSK0001239	4 - Low	Pending	Andree		Release Process Task
RPTSK0001240	4 - Low	Pending	Andree		Release Process Task

4.2 Exercise #5 – Challenge - Add an Approval for the Release Process

If time permits, this Exercise will be discussed interactively during the Hackathon.

5 Integrate your App in the Service Portal

5.1 Exercise #6 – Release Process Request Catalog Item

Was determined to be of minor interest for the Hackathon.
Can be presented during the workshop if desired.

5.2 Exercise #7 – Request a new Task for the Release Process

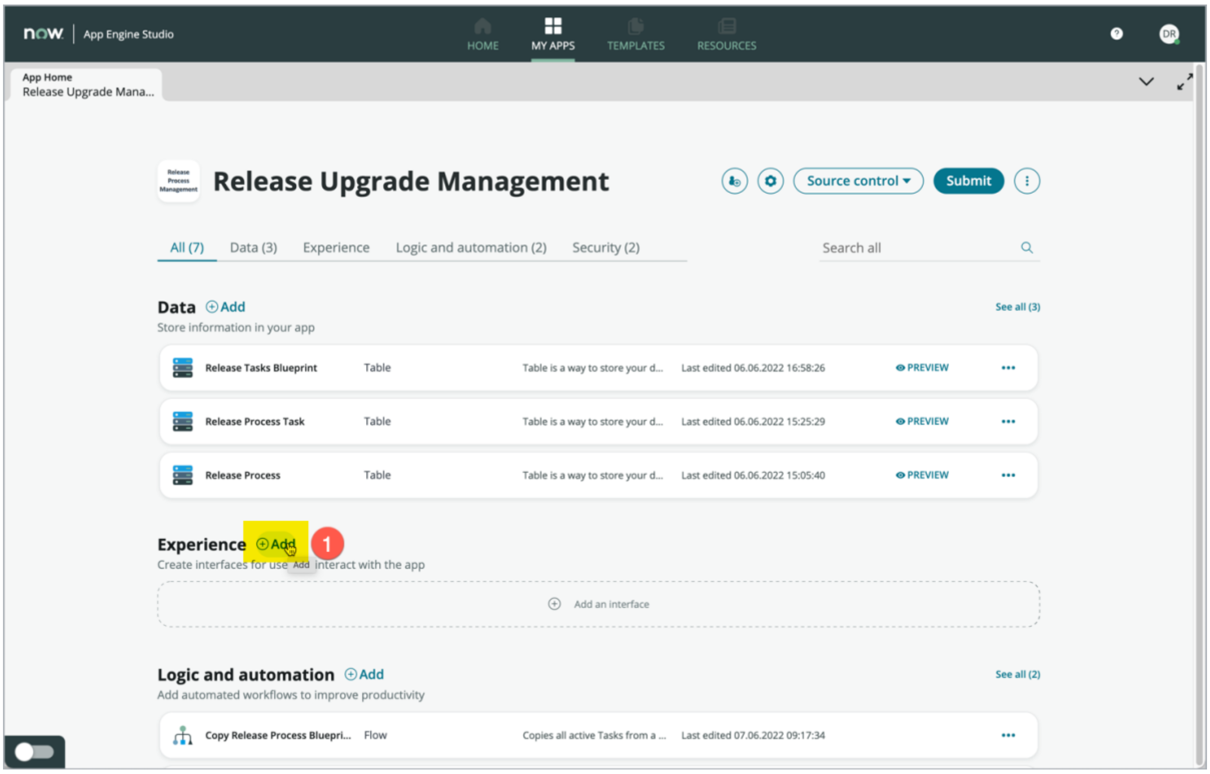
Was determined to be of minor interest for the Hackathon.
Can be presented during the workshop if desired.

6 Create a great Workspace for your App

6.1 Exercise #8 – Create a Workspace for the Release Process App

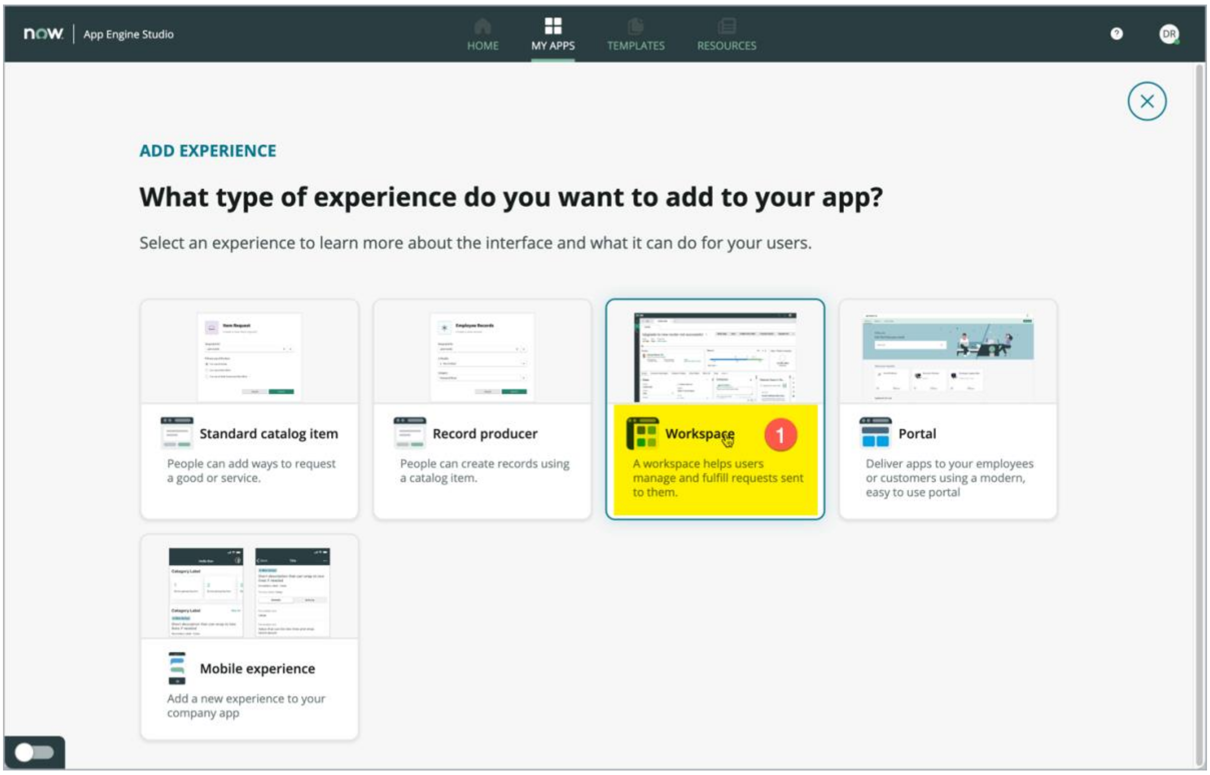
Start the creation of your Workspace from the Dashboard of your App in App Engine Studio.

Click “**Add**” in the “**Experience**” section. (1)

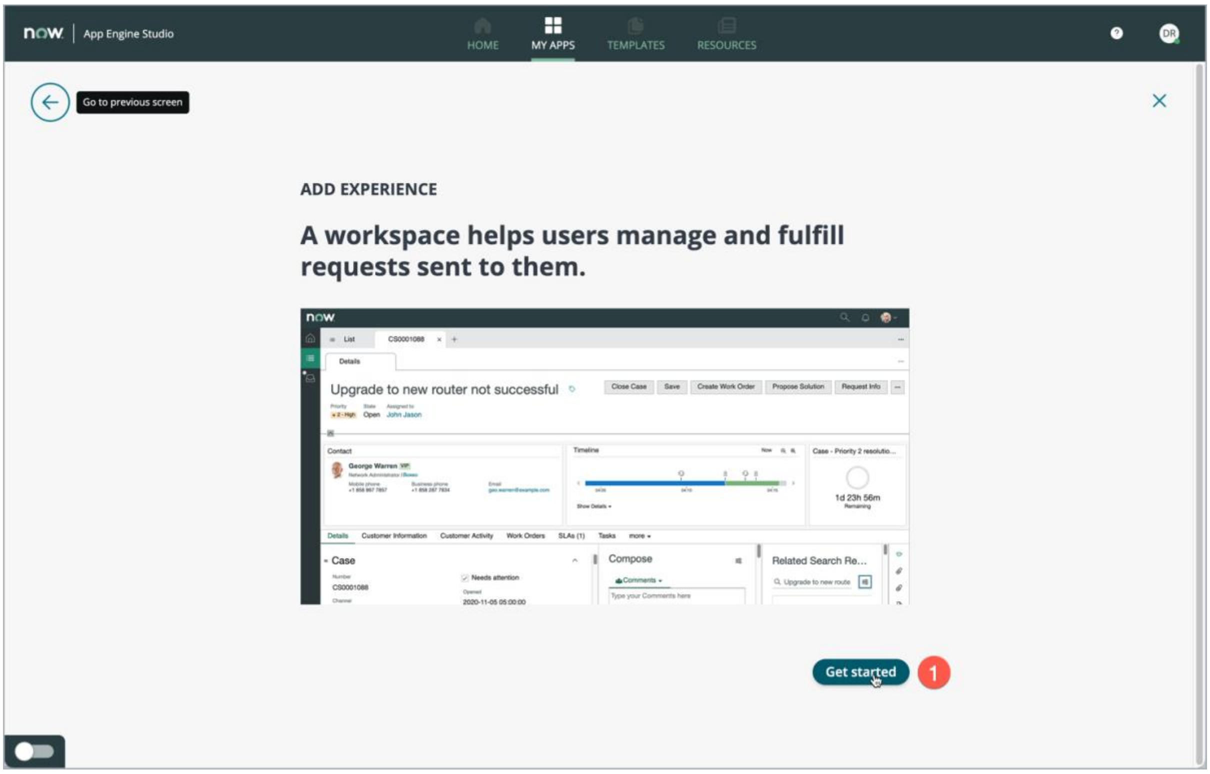


The wizard to create new Experiences (User Interfaces) opens.

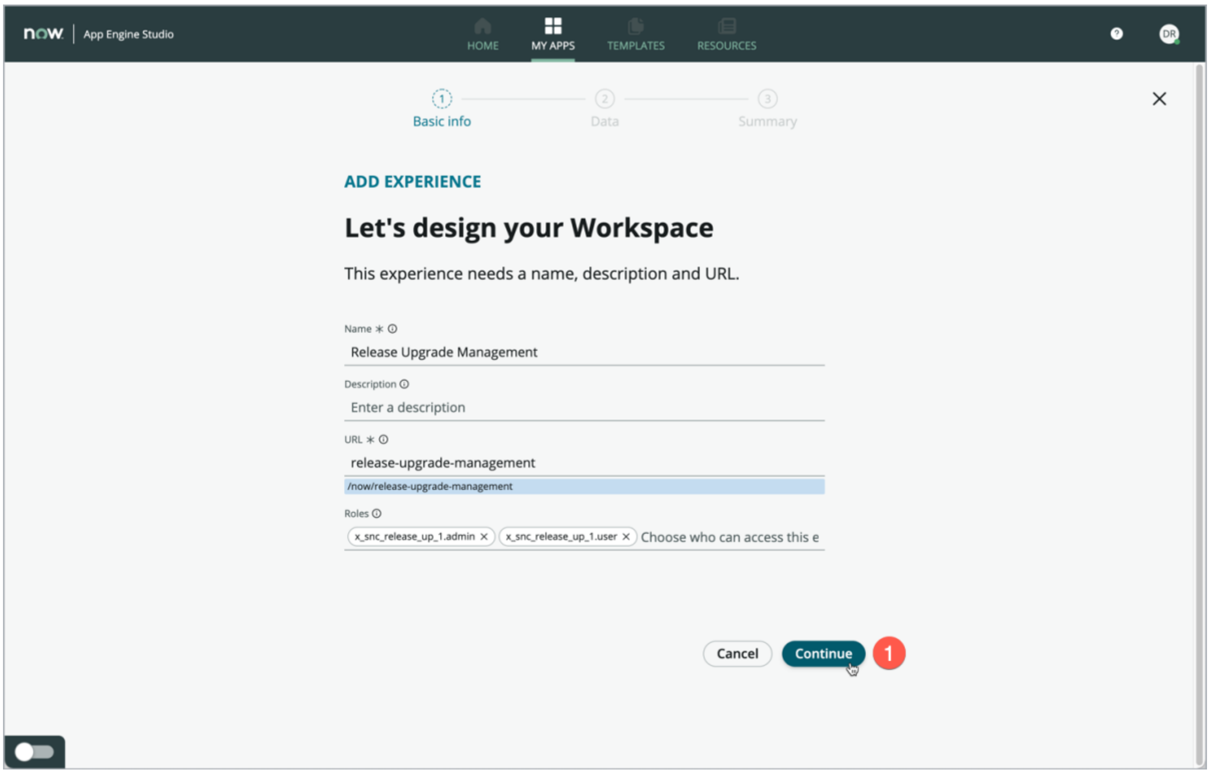
Select the "**Workspace**" (1) card to continue.



Click "**Get started**"

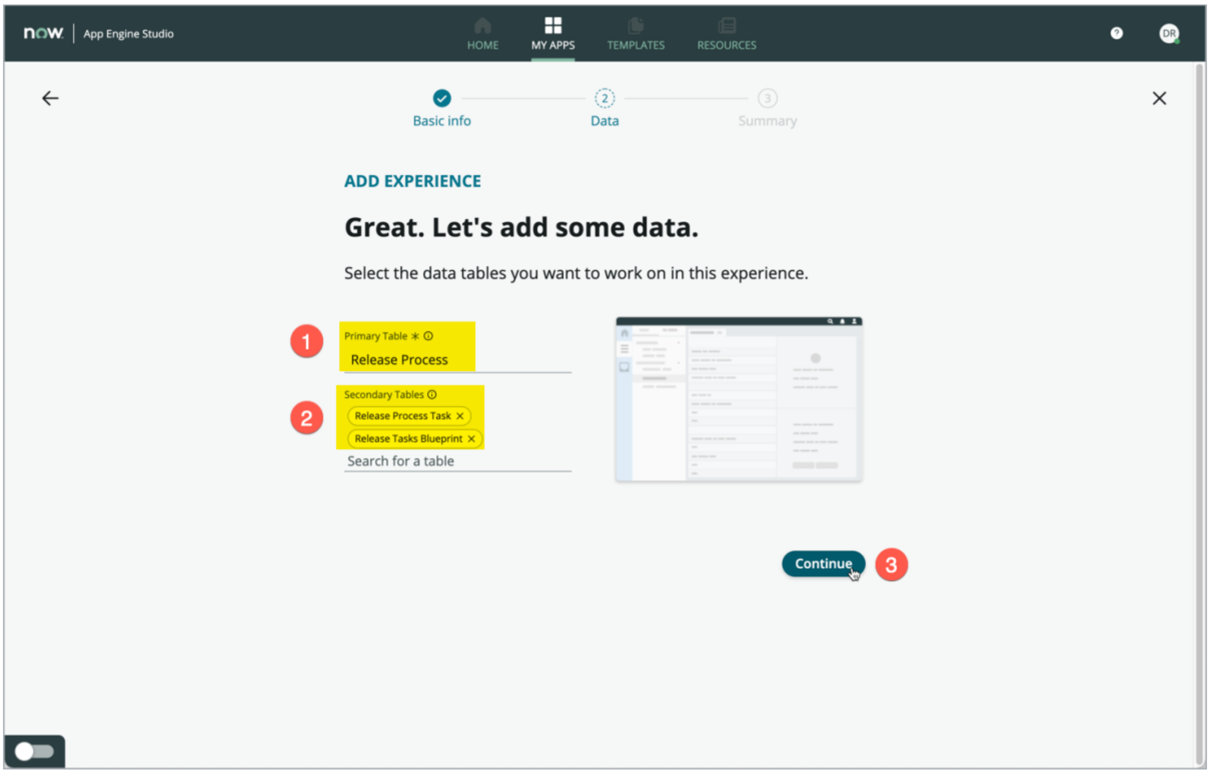


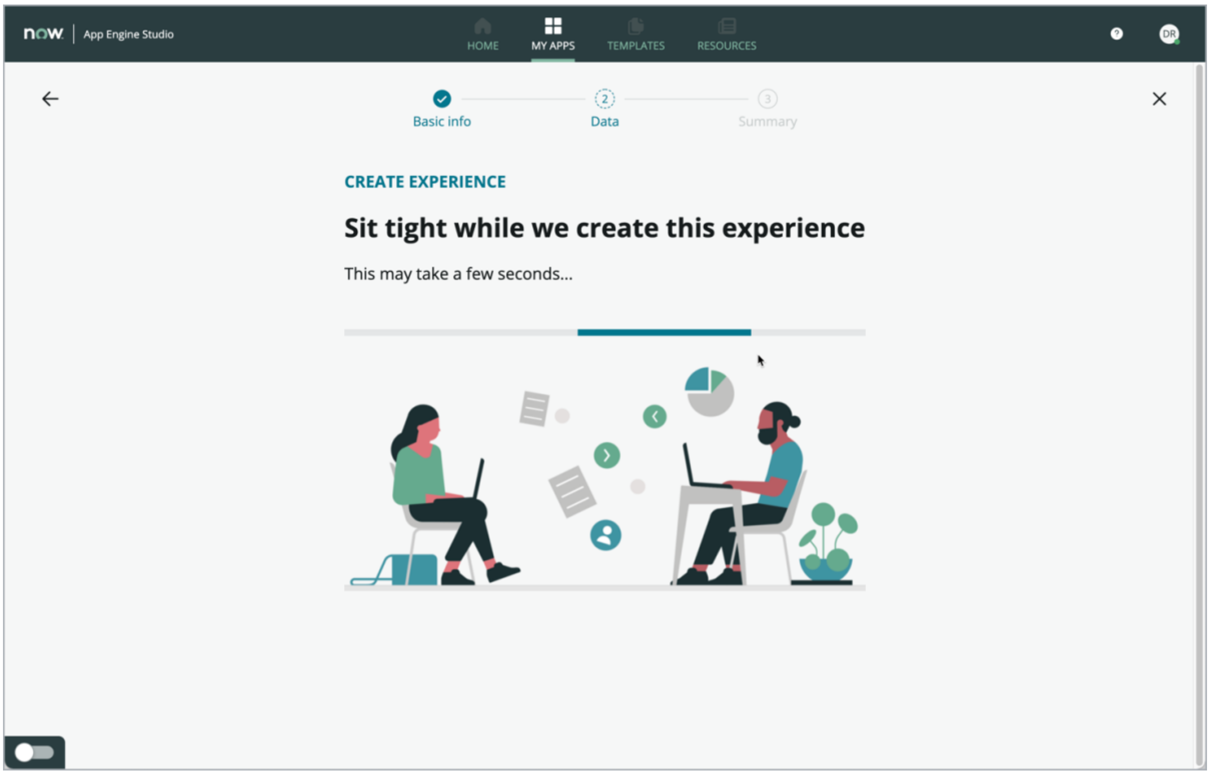
Review the default Name and Description for your Workspace.
Click “**Continue**” (1)



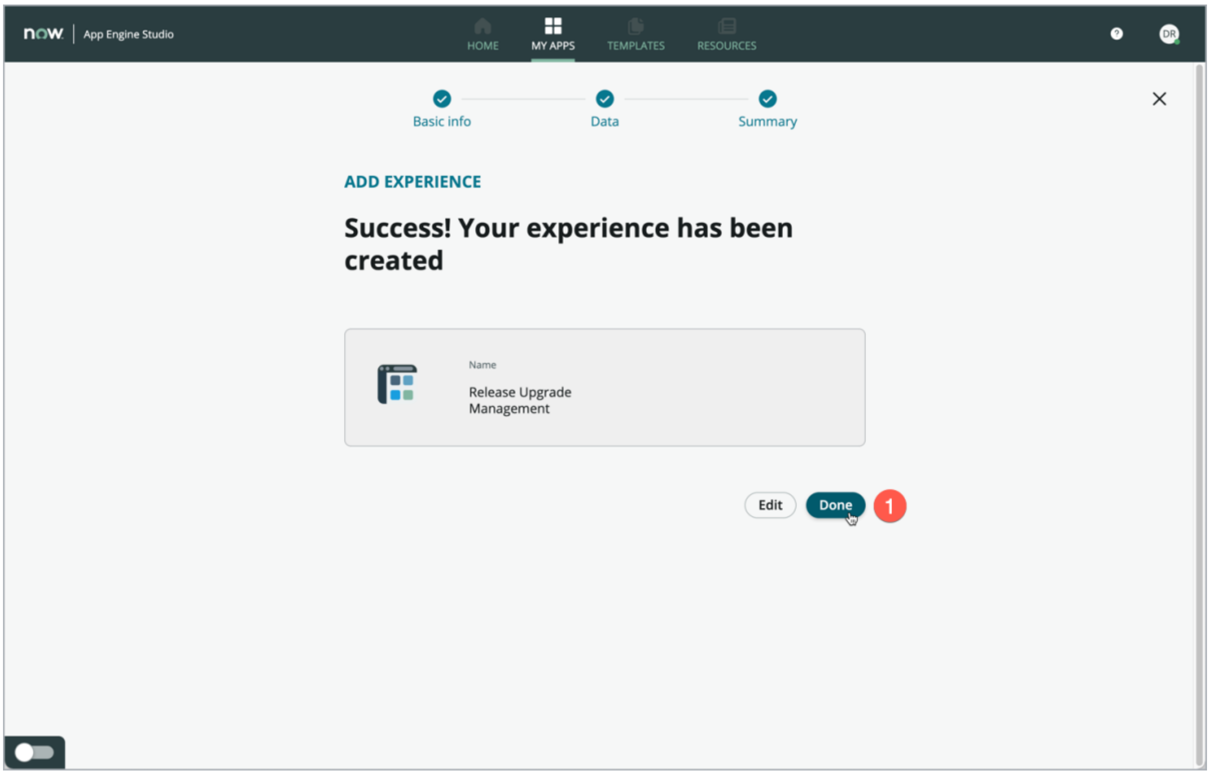
In this step, you can select which tables to set up as Lists in your Workspace.

Make sure, the “Release Process”table is the “Primary Table” (1), and select the “Release Process Task” and “Release Tasks Blueprint” tables as “Secondary Tables” (2).
Click “Continue” (3)

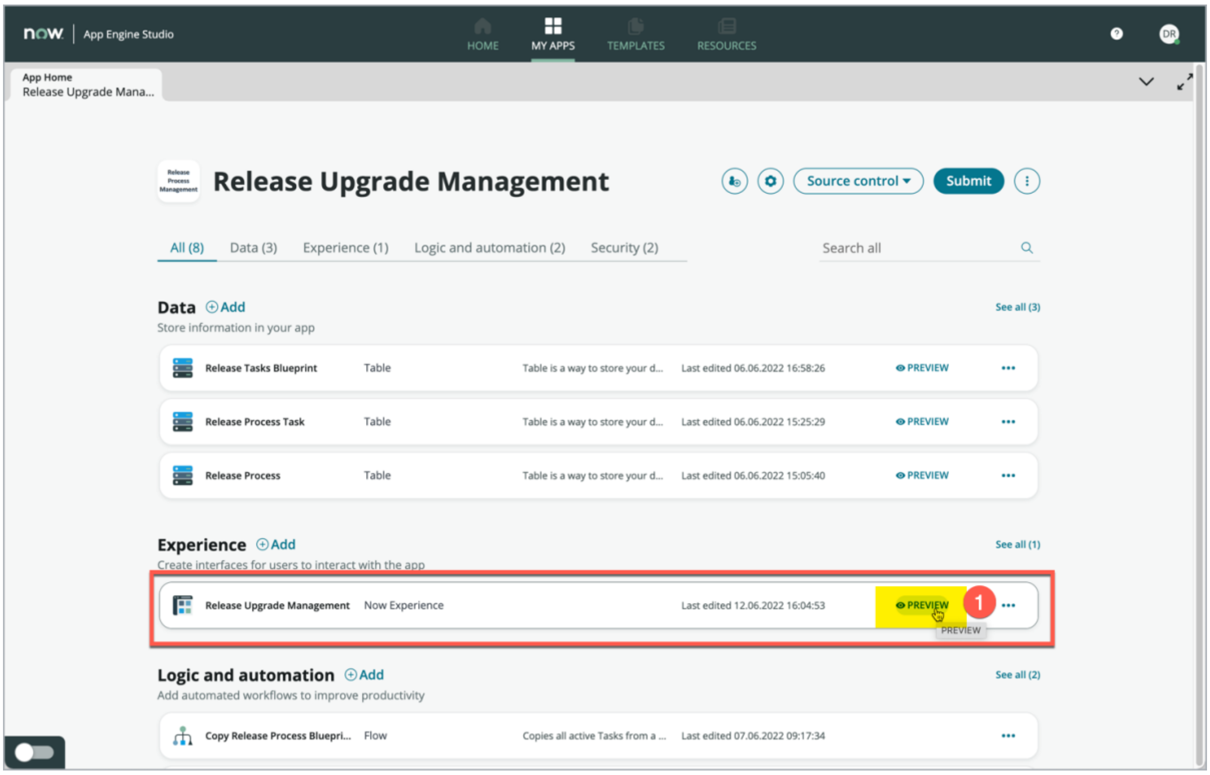




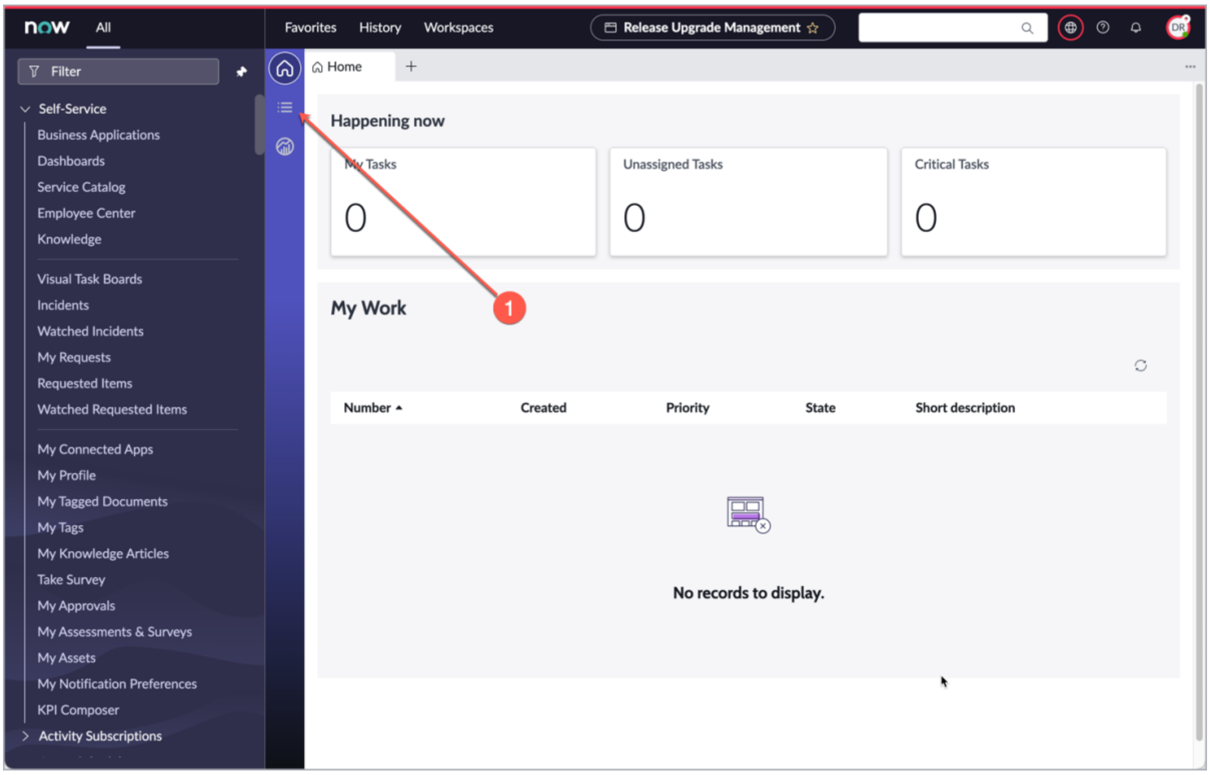
As soon as the wizard finishes, click "Done" (1) to return to your App's Dashboard inside App Engine Studio.



You can find the new Workspace (shown as "Now Experience") in the "Experience" section . Click "PREVIEW" (1) to open a new tab for the Workspace just created.

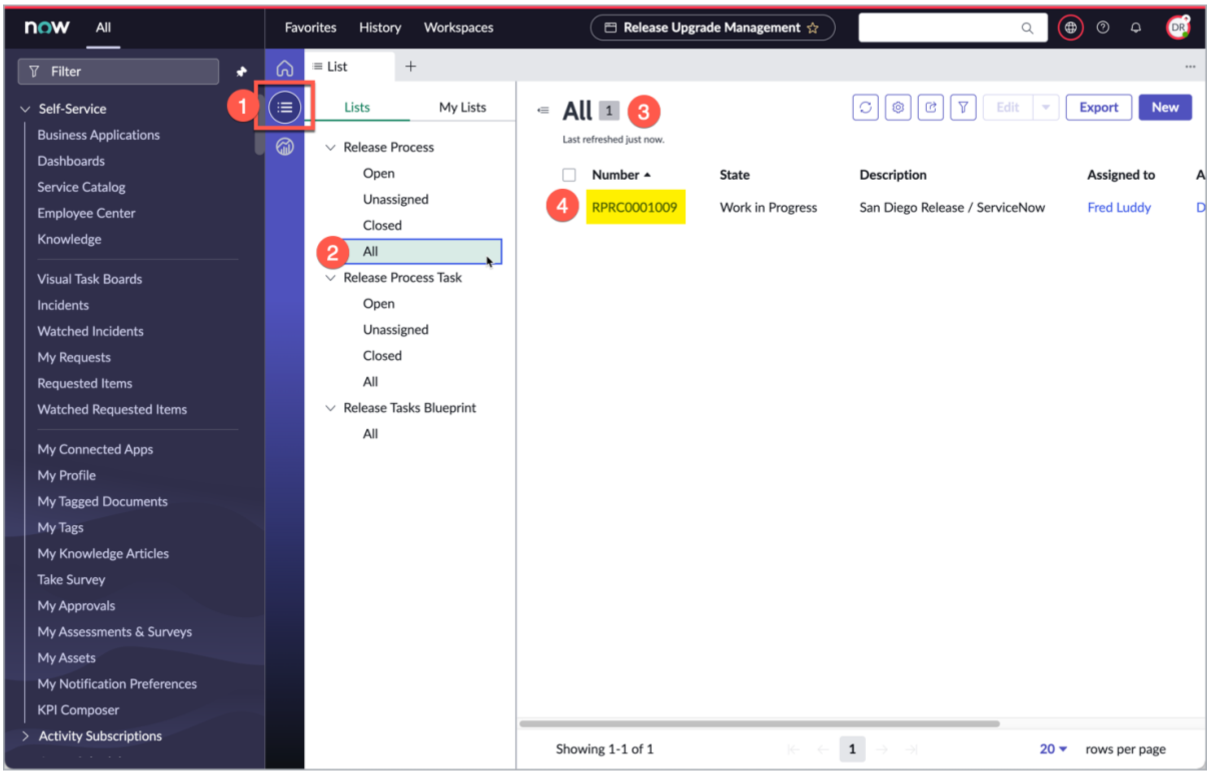


On the Workspace, click the “Lists” icon in the left side navigation panel (1).



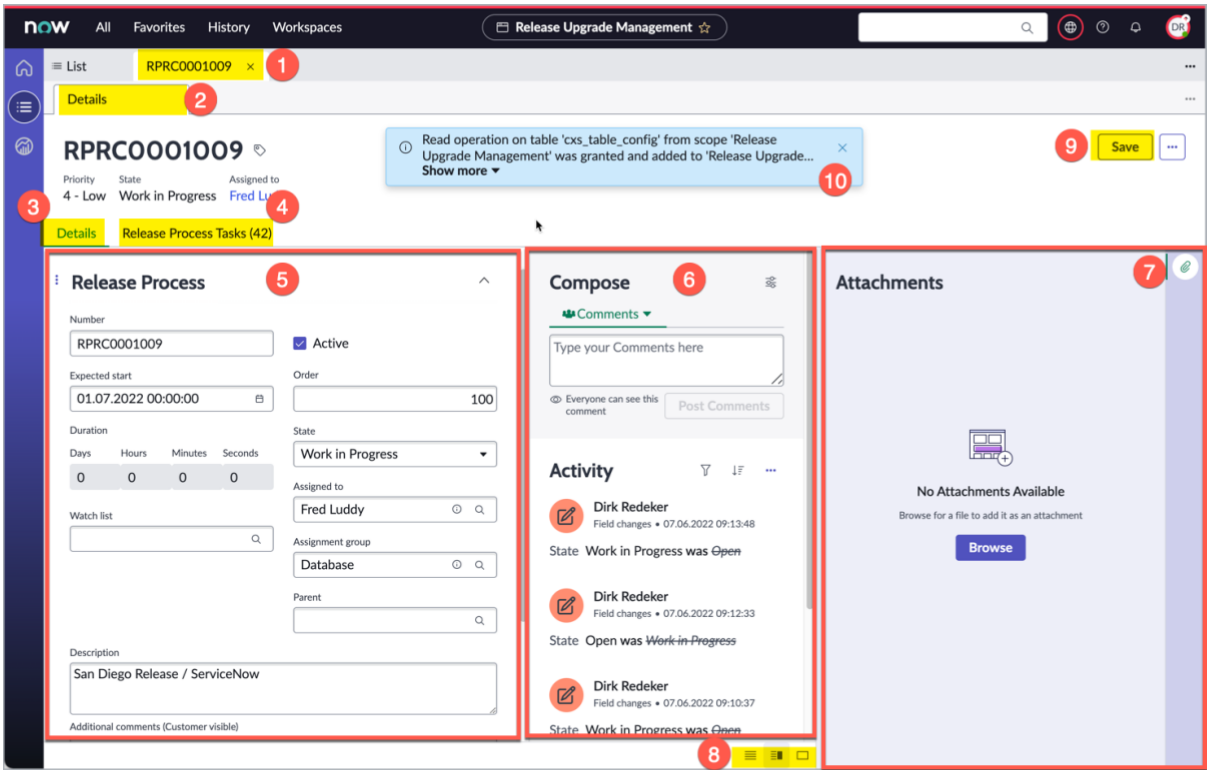
The Lists tree view will be shown and you can select an entry (2) in the tree view to show the corresponding list of records in the right main part (3).

Click the first column of any record (4) in the list to open a new tab inside your Workspace showing the details of your record.

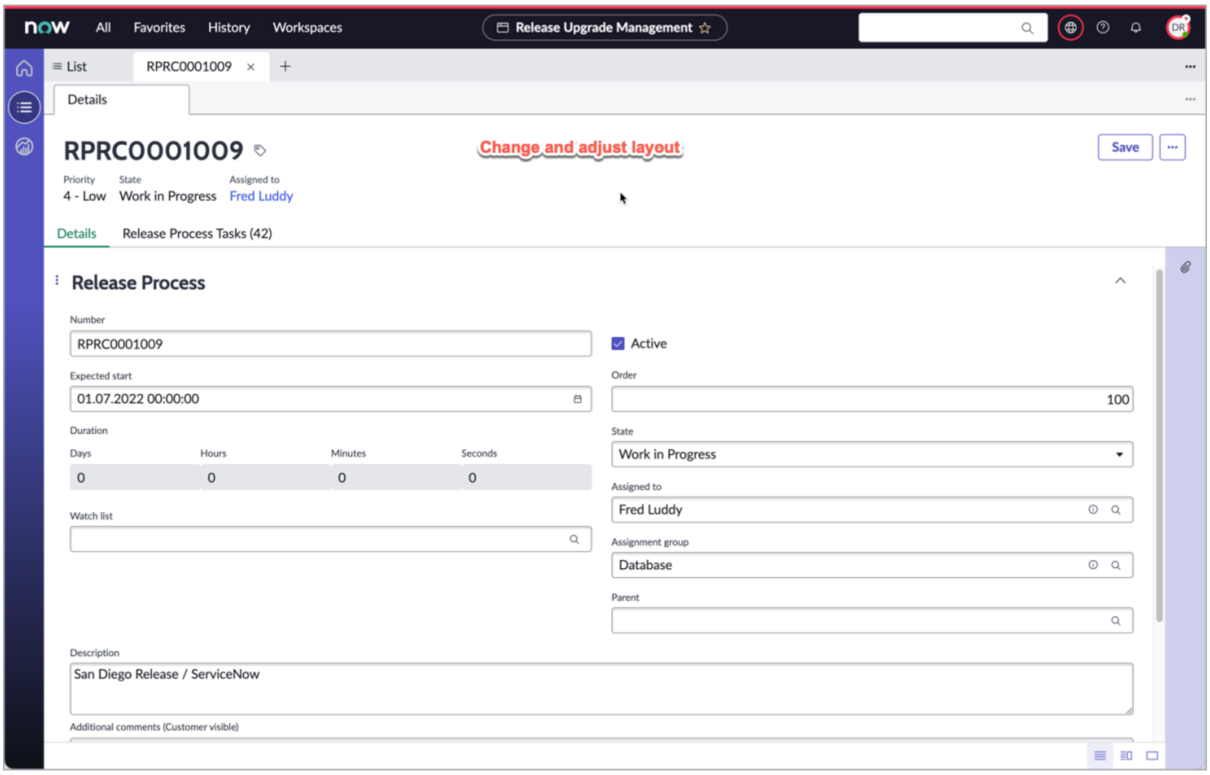


The workspace is build in some hierarchical structure.

- In the top row (1), you can find any record opened from the lists tree view.
- Each Record opened from the lists tree view can have one or more tabs (2) with records relating to the current “main” record.
- By default, one tab (3) is opened (“Details”) showing the Form view of the Record selected.
- The Related Lists (4) defined in the Classic UI are also shown on separate Tabs for the current record.
- The Details of the main Record are shown in the left part of the screen (5), where all Form sections are arranged in vertical order
- The “Compose” pane (6) shows the Activity log of the Record. This panel is only available, when you added the Activity Formatter to the Form View in the Platform UI. Users can review the change history of the record, but also can add “Comments” and “Work notes” here.
- If there are Attachments available for this record, they will be listed in the right part. But the right panel (the “Contextual Side Panel”) can hold different Information, which can s´be configured and selected from the side navigator at the right side (7).
- Change the layout of the form using the buttons at the bottom (8)



You can switch the way, the form is shown, like e.g. shown in the screenshot below.

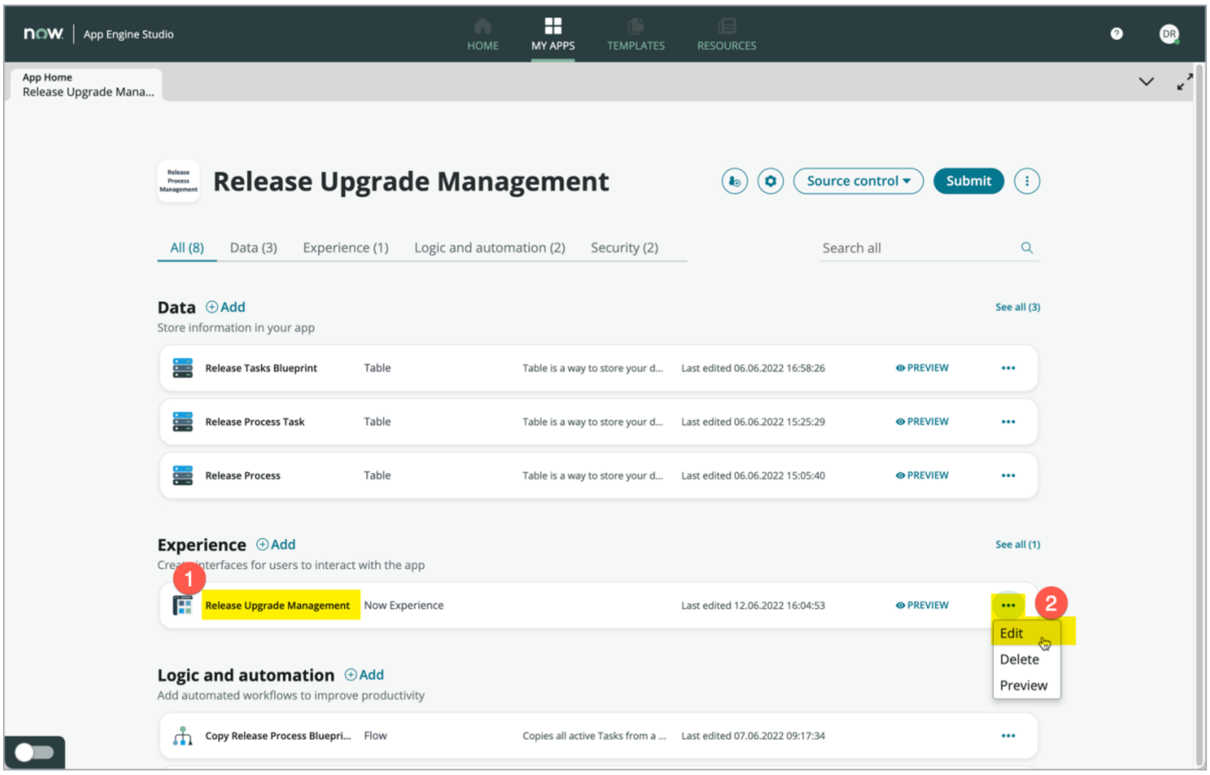


6.2 Exercise #9 – Configure the Landing page to a nice dashboard

Start from the Dashboard of your App

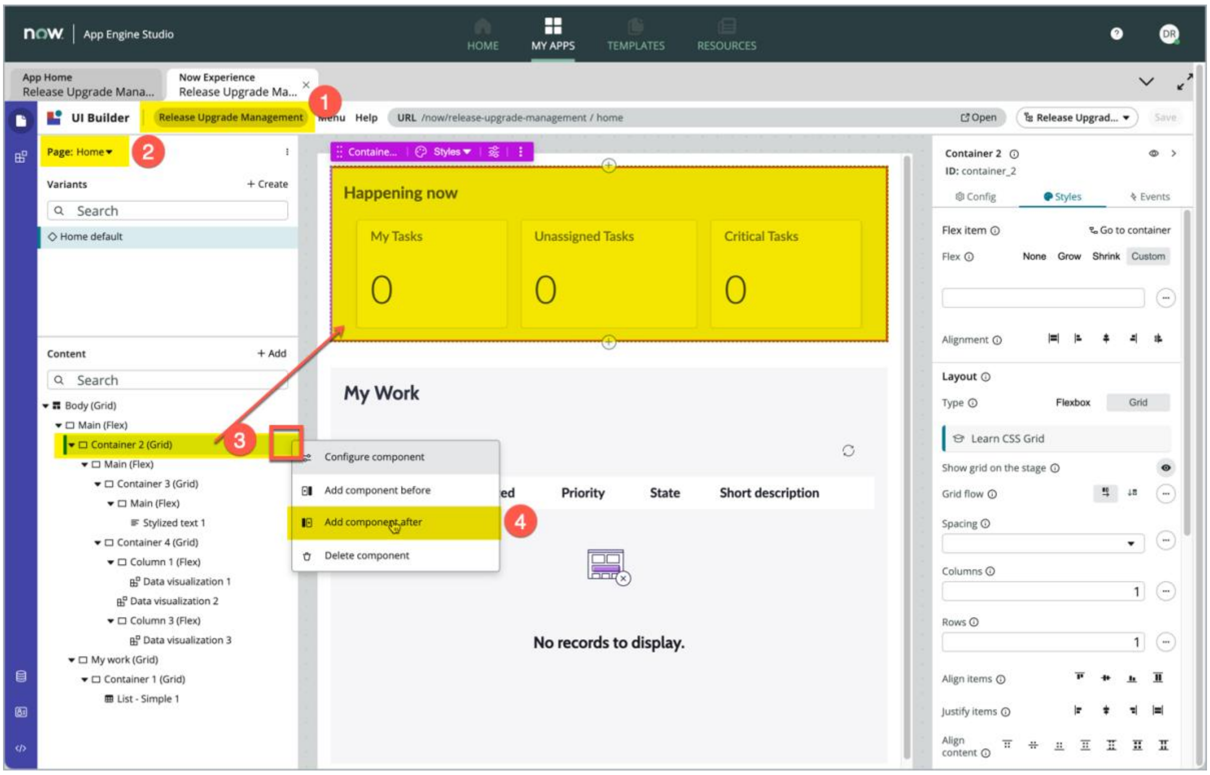
Open the Workspace in UI Builder inside App Engine Studio by

(1) Clicking the name of the Workspace, OR
(2) Clicking the dot-menu on the right side and select "Edit"

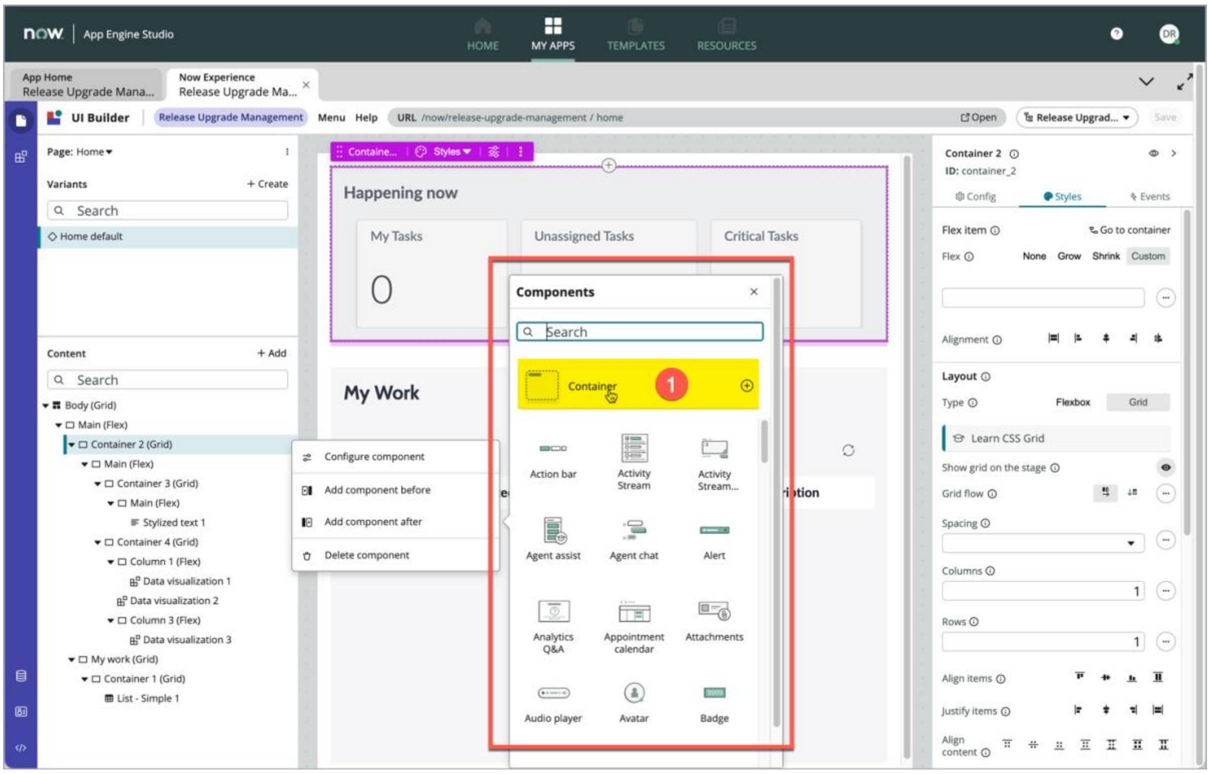


UI Builder opens with the Now Experience ("Release Upgrade Management") Landing Page – here "Home" (2).
The left panel (3) shows the structure of the Screen in the middle.

Click the three dots menu at the end of the line "Container 2 (Grid)" – see screenshot below, and select "Add component after" (4)

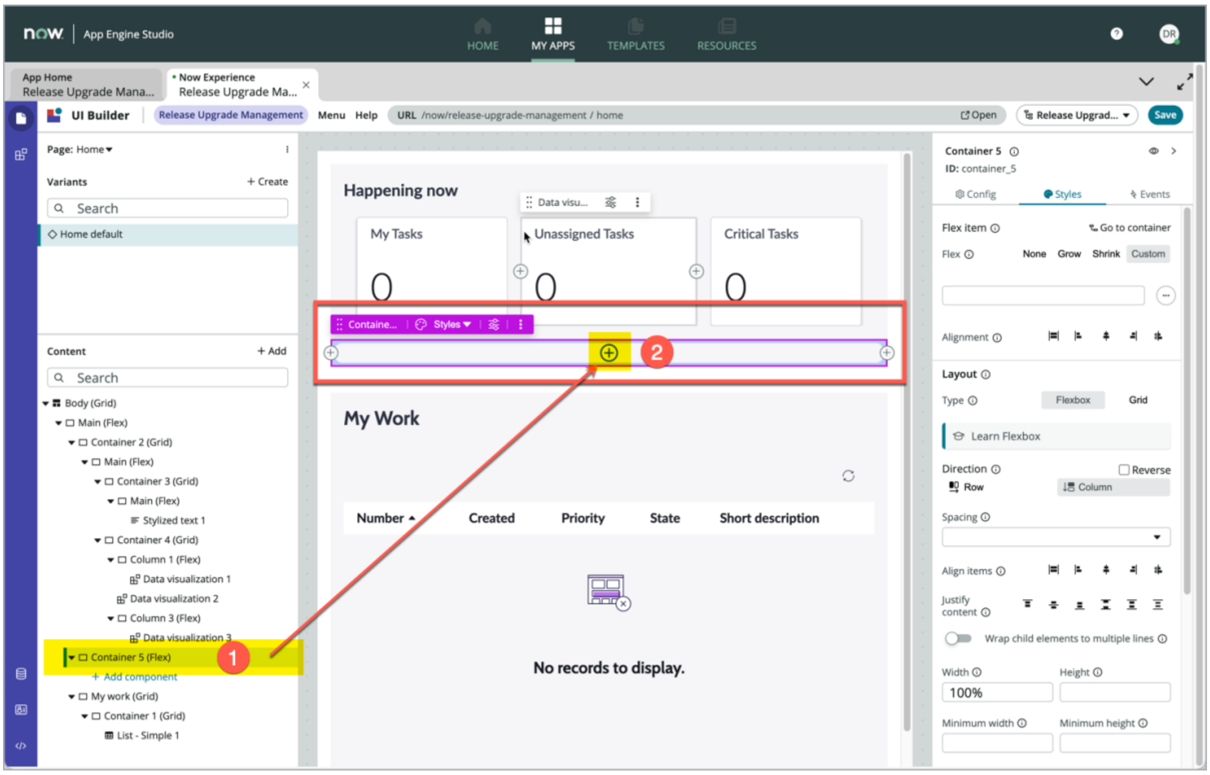


A popup dialog opens, where you can select, which component to add to your screen. Select the “Container” Component (1).

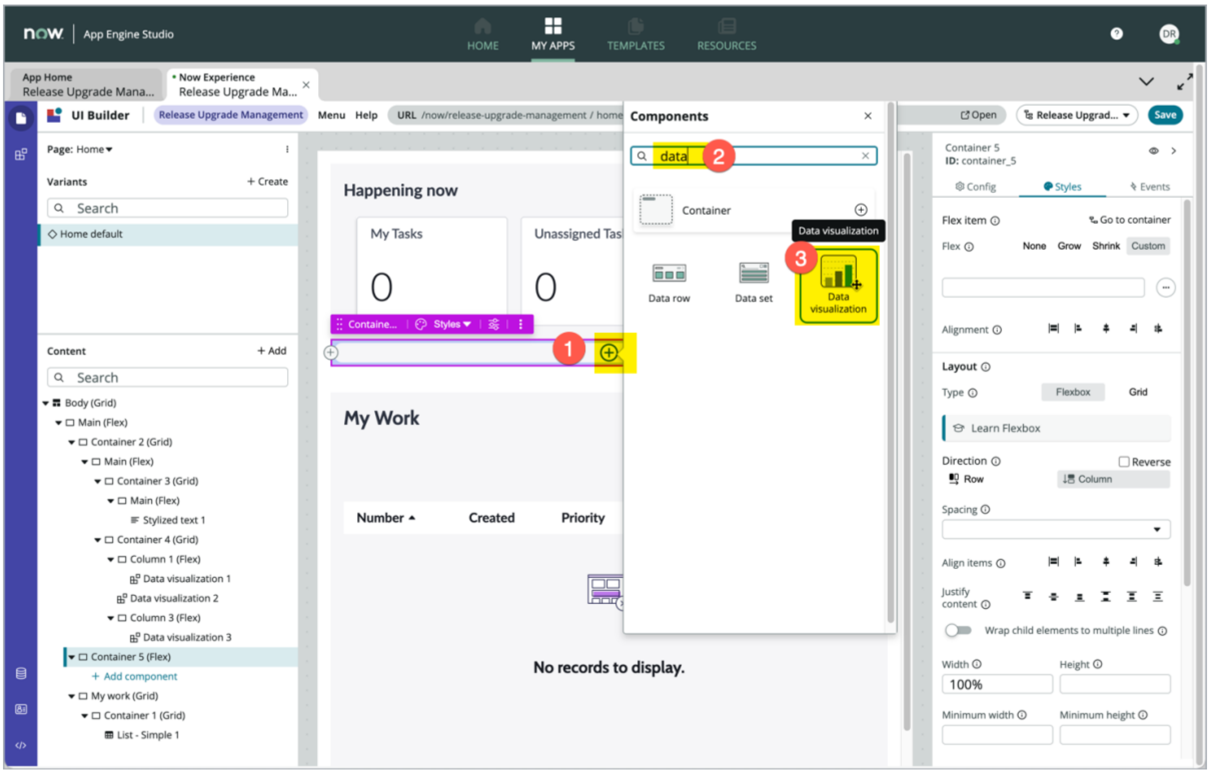


This adds “Container 5 (Flex)” to your page structure (1) and also shows an empty area on the page canvas in the middle of the screen (2).

Click the plus (+) sign (2) inside the new container to add a component INSIDE of that container.

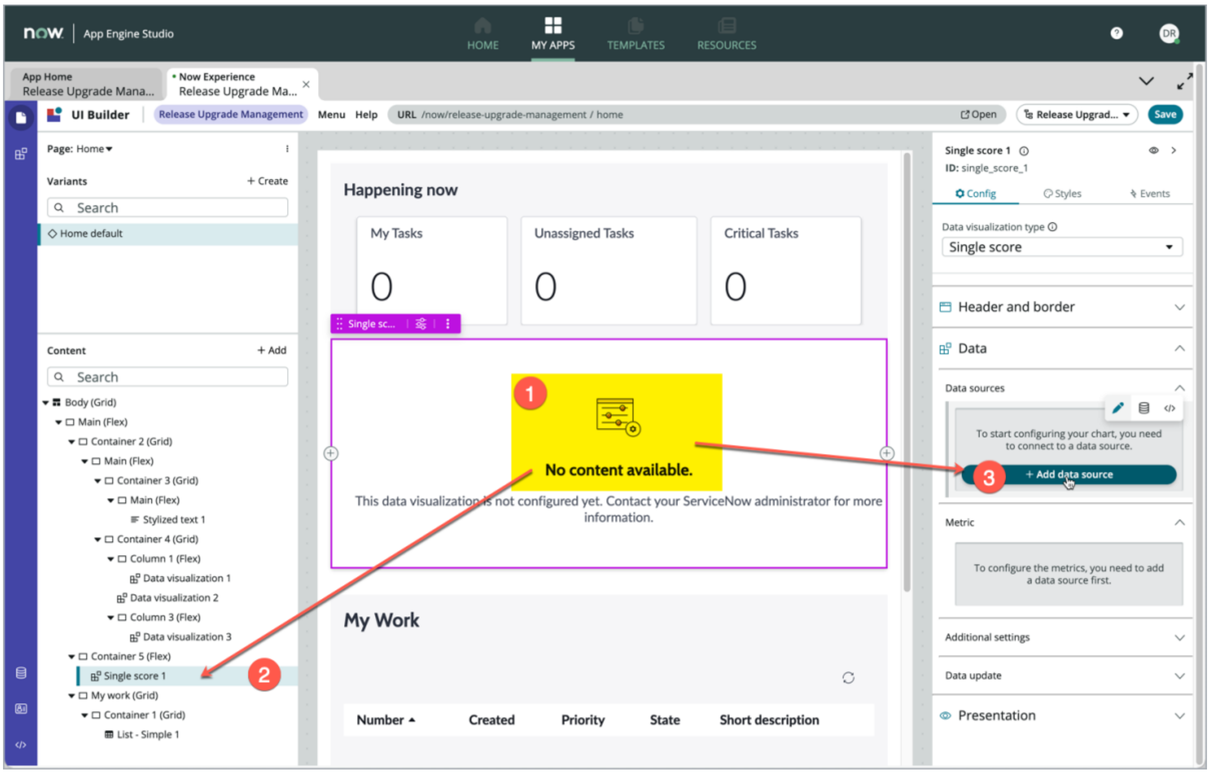


Again, the Popup dialog appears. Type “data” in the search field (2) at the top of the popup and find the “Data visualization” Component (3). Click that Icon (3) to insert it into the Container.



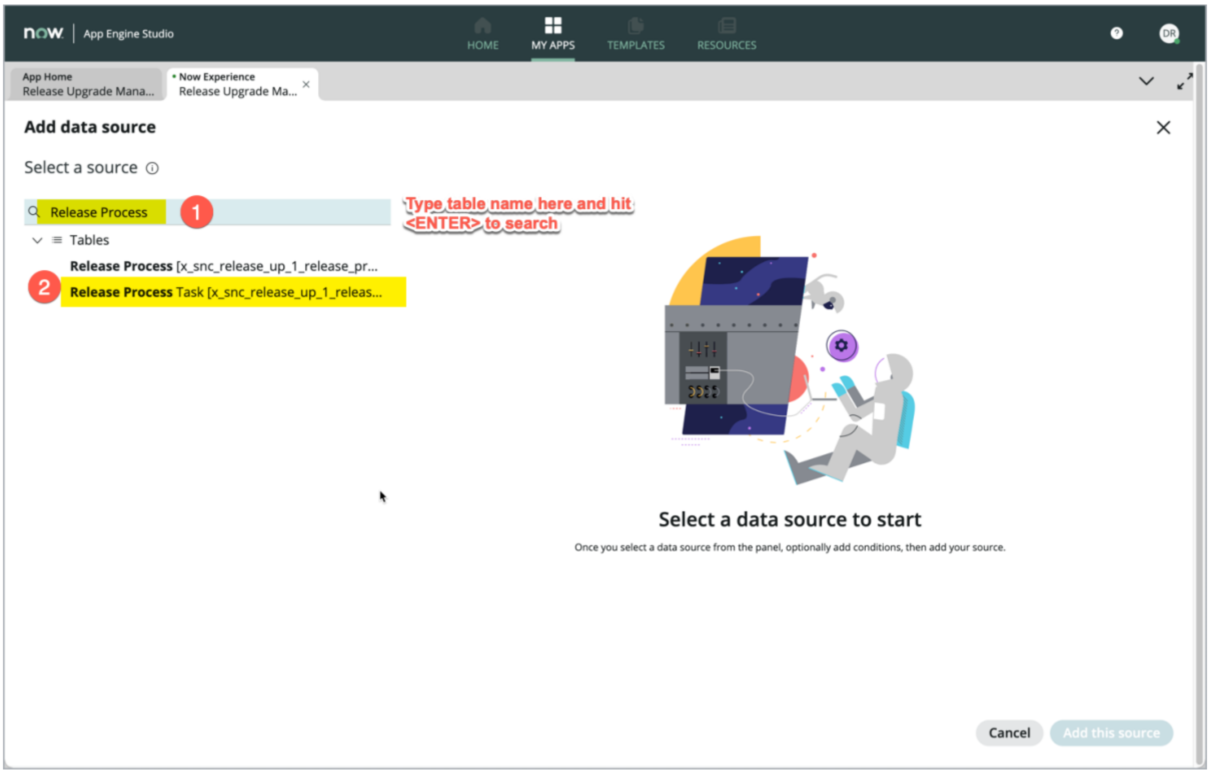
Now, you can see the “Single score 1” added to the content tree (2) and also an empty “Data Visualization” component on your page (1).

Click “+ Add data source” (3) in the right properties panel.



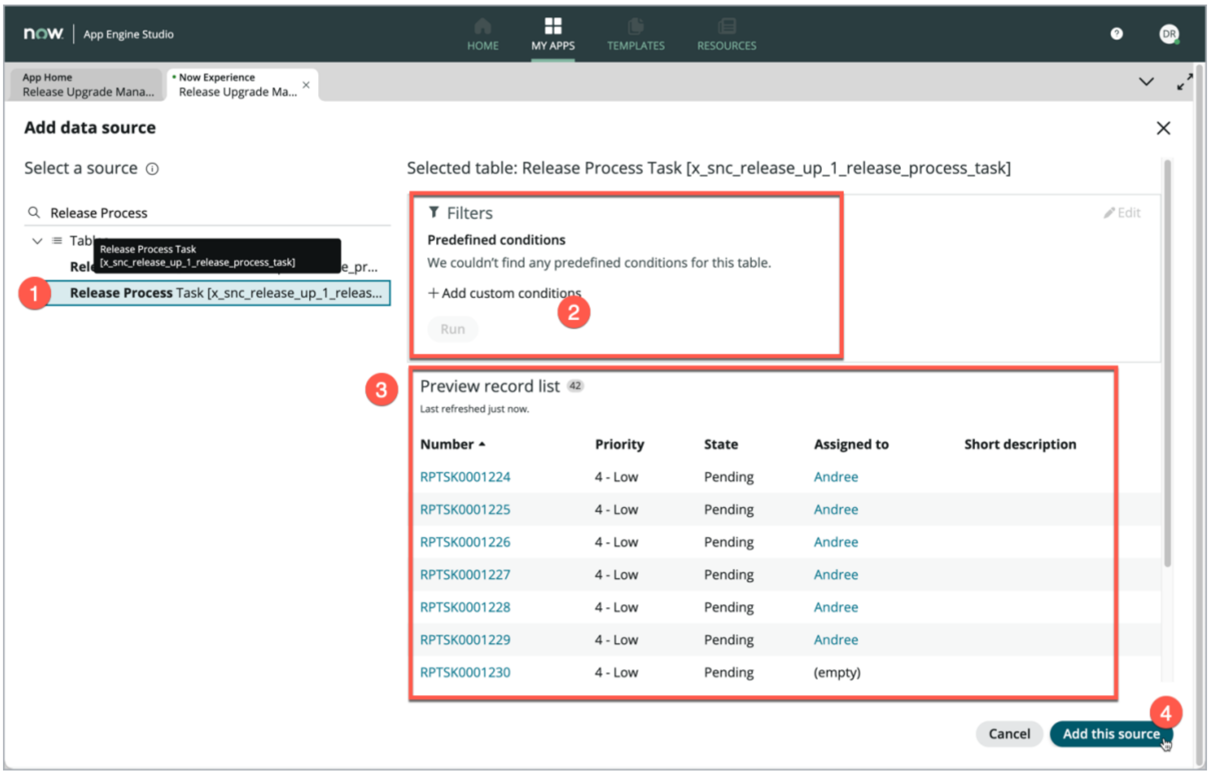
An overlay window opens where you can easily select, which data to show in your data visualization. Type the name of your table into the search field (1) and press <ENTER>.

Select your data table (here the "Release Process Task") form the search results (2) below.



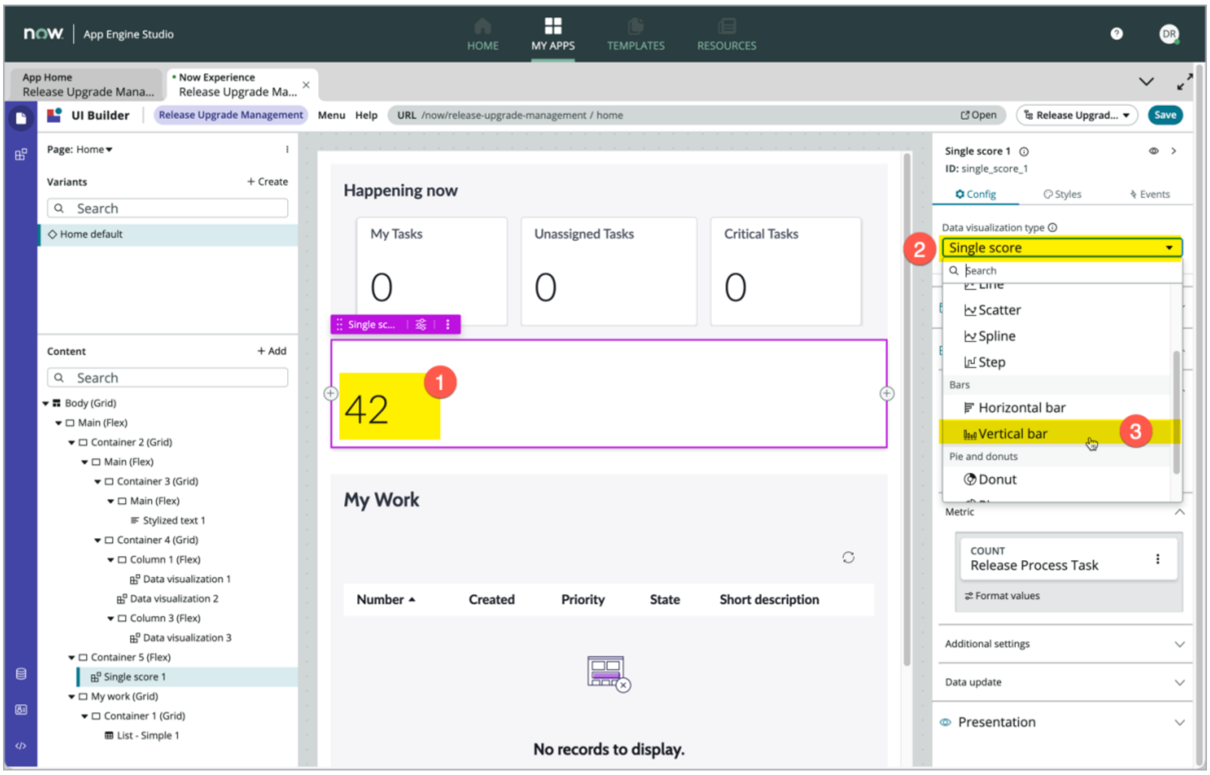
The Filter area (2) appreas, where you can configure a filter for the data used to fill the "Data Visualiozation" component. Do not add any filter yet. You find a preview of the data below the Filter (3).

Click "Add this source" (4) to close the overlay window and select the data source.



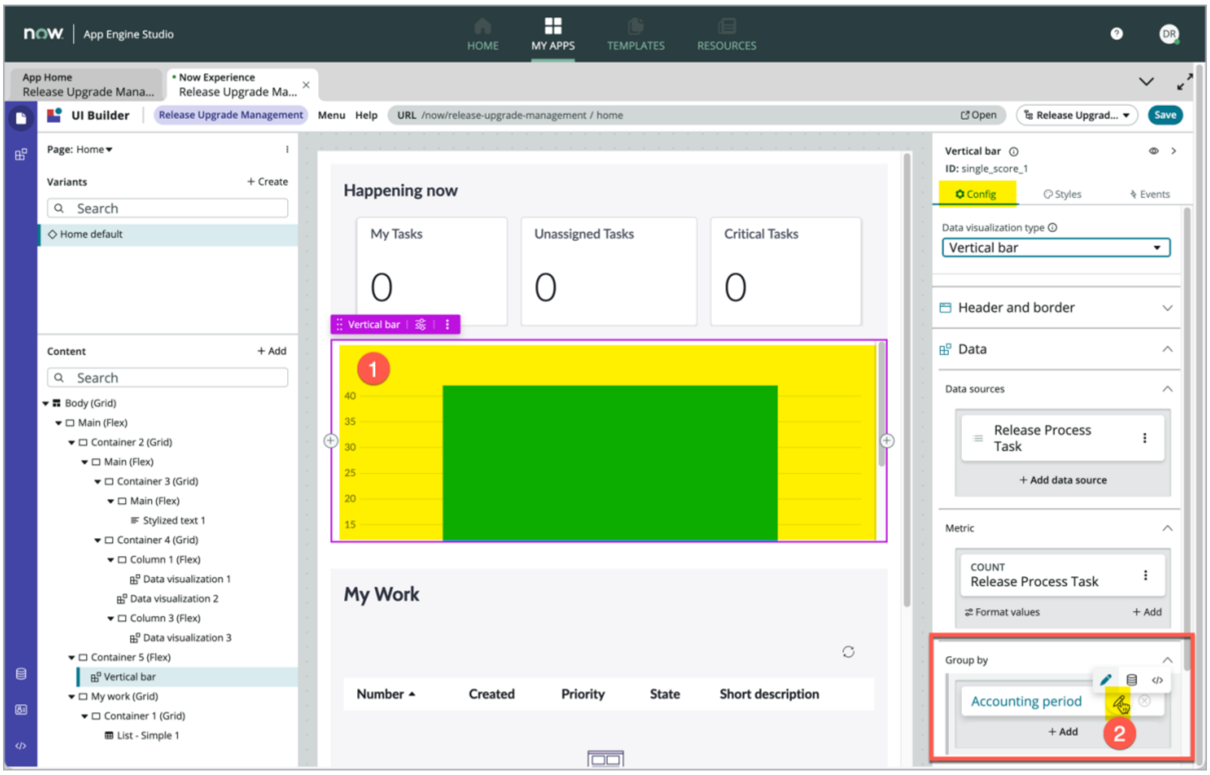
The value shown in UI builder immediately changes Data Visualization (1) shown in the design canvas in the middle. This is a WYSIWYG style of designing your pages.

As the default “Data visualization type” (2) is “Single score”, open the dropdown and change it to “Vertical bar” (3).



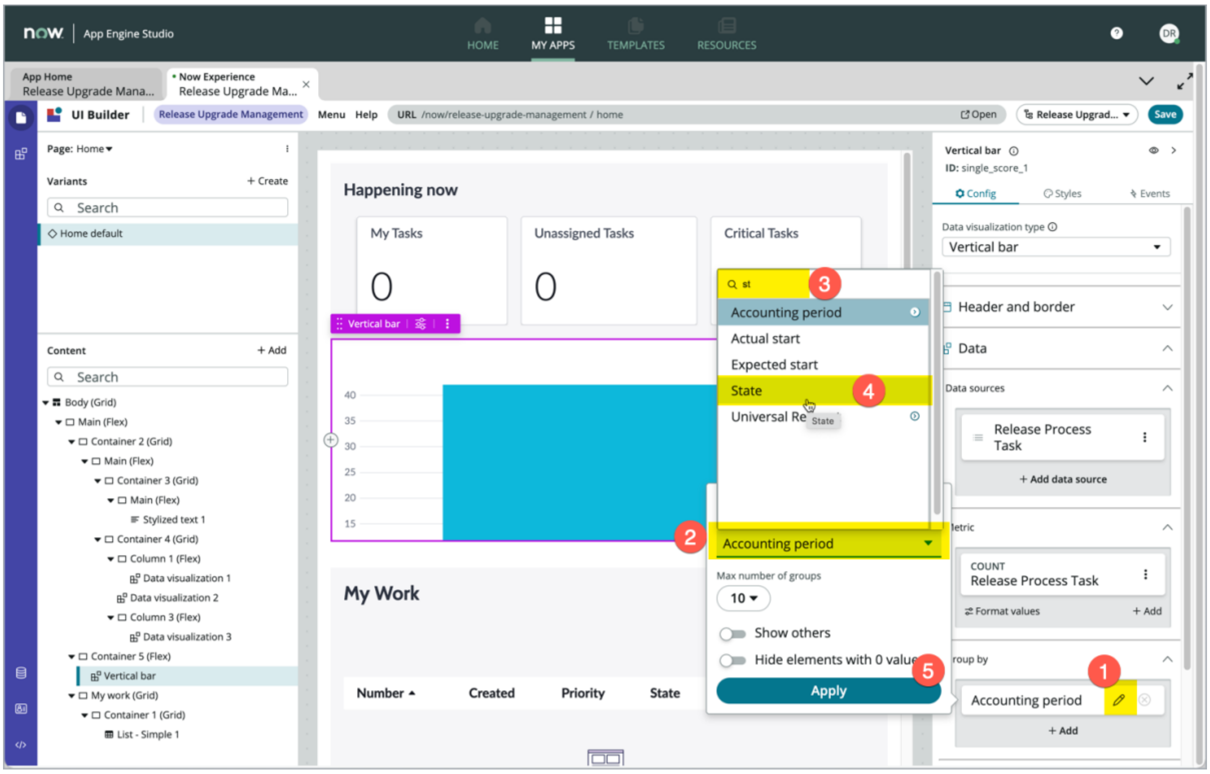
Review the change of the appearance from the Component (1), which happens instantly.

Scroll down in the right panel and find the "Group by" field. Click the Edit icon (2) to open the popup window for configuring the way, how the bar chart component groups records to build it's bars.

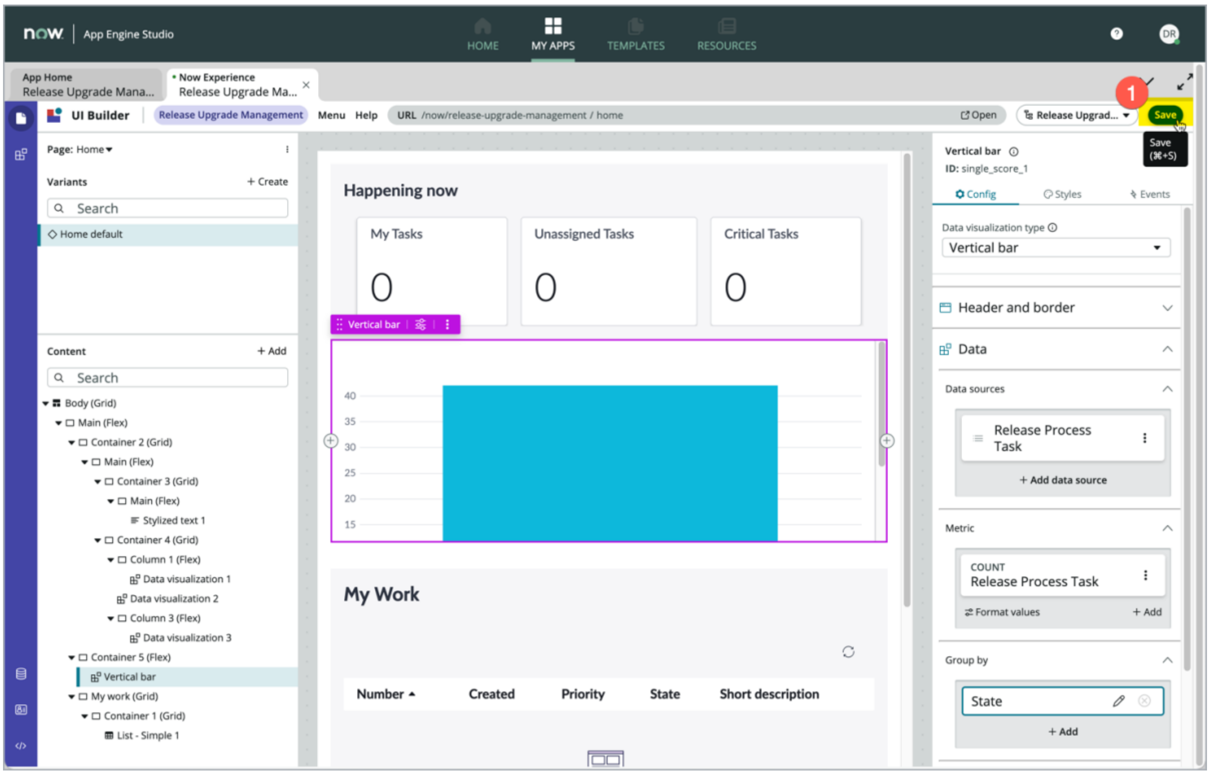


Click the Dropdown Box (2) to select a different field for the grouping of the bar chart. Type “St” in the search field (3) to quickly find and select the “State” field (4).

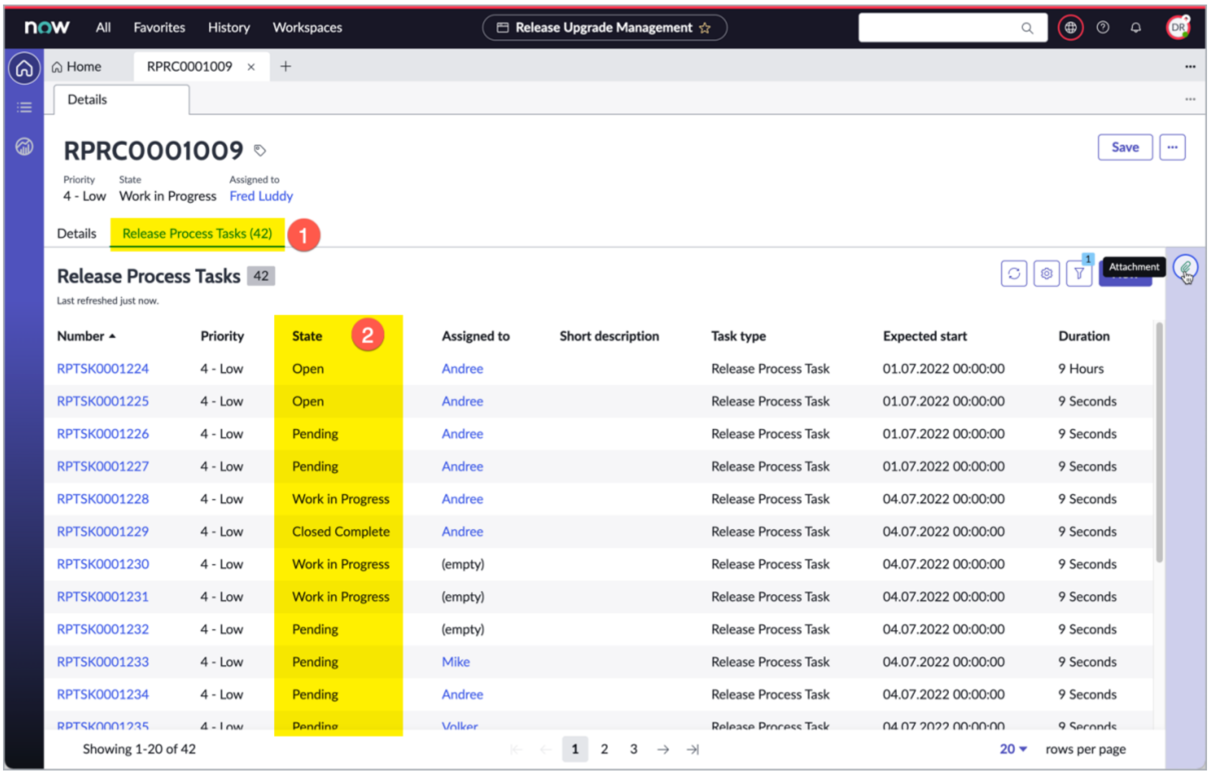
Click “Apply” (5)to close the popup and use the settings selected.



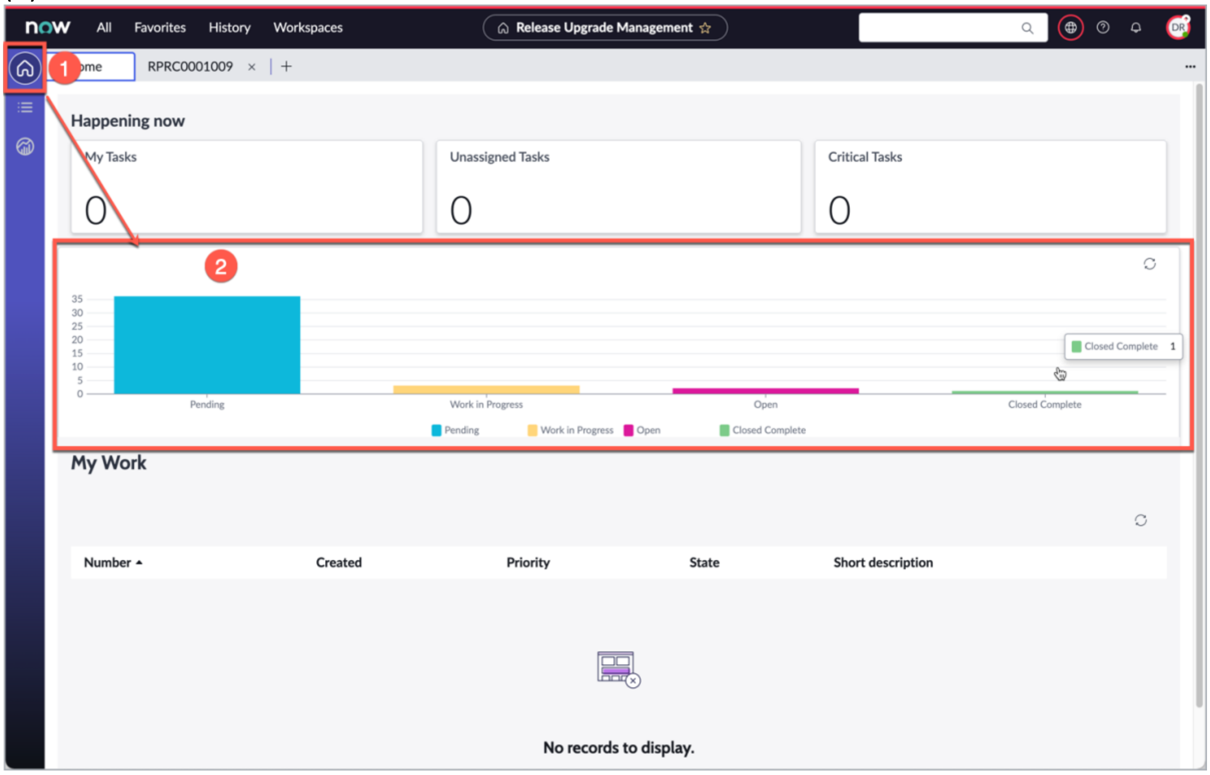
Click “Save” in the upper-right corner (1) to make all your changes permanent.



Go back to your workspace and switch to the Related List tab “Release Process Tasks” (1). Manually change some of the “Release Process Tasks” **State** values (2) – see screenshot below.



Return to your Landing page (1) of your Workspace and review the results of the changes (2).

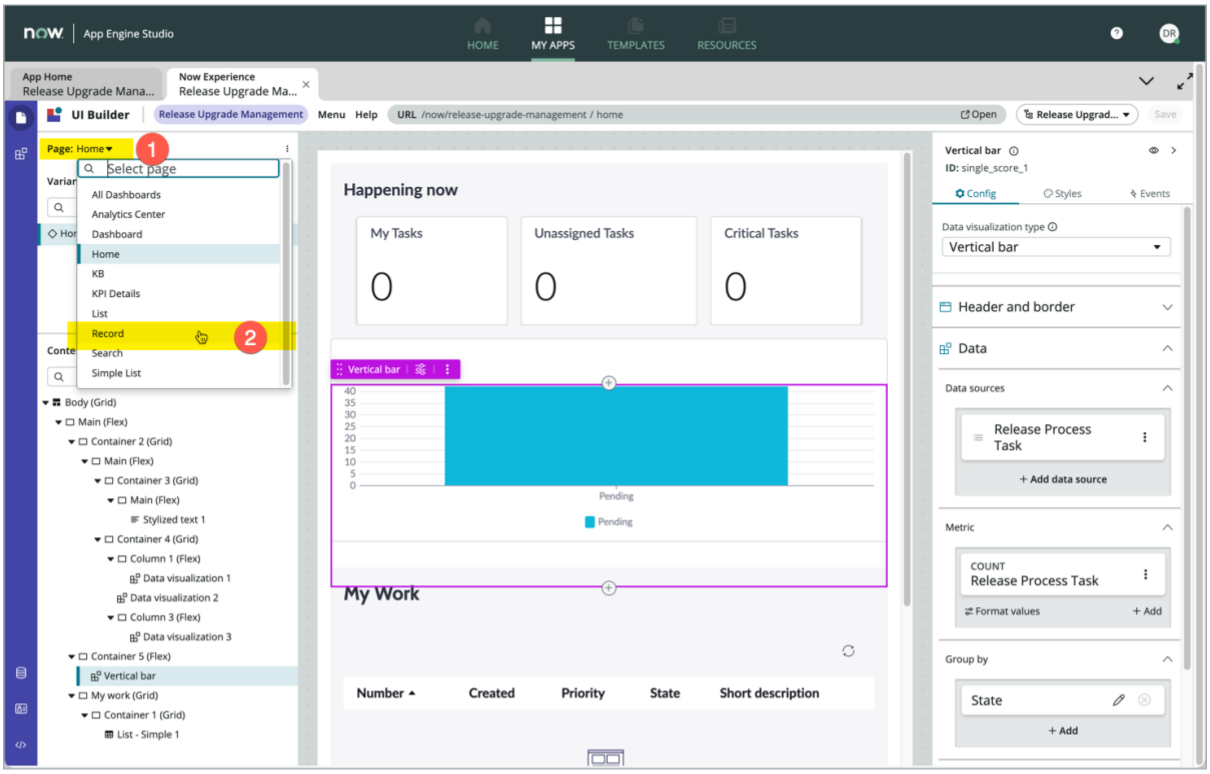


CHALLENGE:

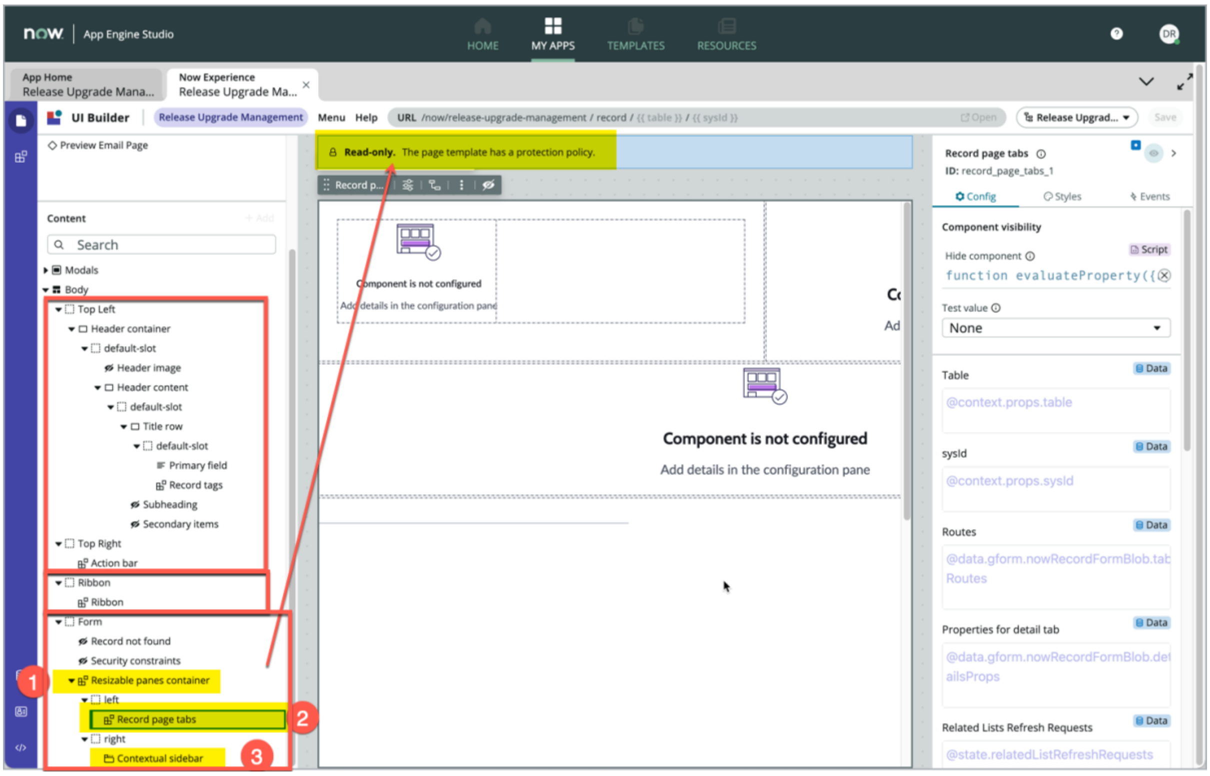
- Add more diagrams to your Landing/Home Page.
- Change more Layout-Options for your new Chart (e.g. Set a Title for the chart)

6.3 Exercise #10 - Review the “Form” Page of the Workspace

In UI Builder, select the “Record” page (2) in the Page dropdown list (1).



Review the layout tree for the "Record" page, especially the "Resizable panes container" (1), the "Record page tabs" (2) and the "Contextual sidebar" (3)



Context for the Form and the Contextual Sidebar can be configured as well