# **Paging Grids Exercises**

**Exercise 1: Paging Grids**

In this exercise you will create a static paging grid as well as a selectable one using data in a CDT. You will have to add some test data into the rule input yourself in order to have anything display in the grid. Remember, when using more complicated expressions, a lot of the work is already done for you. For extra help, take a look at the SAIL Recipes in the documentation.

<https://forum.appian.com/suite/help/16.1/SAIL_Recipes.html#Display_Array_of_Data_in_a_Grid>

<https://forum.appian.com/suite/help/16.1/SAIL_Recipes.html#Select_Rows_in_a_Grid>

<https://forum.appian.com/suite/help/16.1/SAIL_Recipes.html#Format_Data_from_a_Record_in_a_Grid>

Using a grid field, build an interface that displays requests that have been made already. The interface should adhere to the following guidelines:

1. Create a read only paging grid
   1. Create a rule input for requests and add 5 requests with populated data to display in the grid, including the ids.
   2. Use a gridField to build the grid data.
   3. Load the pagingInfo into a local variable, as that could change if the user clicks to another page or wants to sort the grid.
   4. Use with() to transform the rule input data into a data subset.
   5. The grid should display all 5 fields from the CDT.
   6. Sort the grid by the dateNeeded and show a batch size of 3.
2. Create a selectable paging grid
   1. Add a rule input for the selected requests to store the ids
   2. Update the pagingInfo to a!gridSelection
   3. Add selection and identifiers parameters to the grid field
   4. Update the value and saveInto to reflect the gridSelection instead of the pagingInfo
   5. Save the selected ids to the new rule input

STRETCH GOAL: Add formatting to the grid fields:

1. Update the part to not show the id, but rather the name of the part (hint: take a look at apply() and partial evaluation)
2. Update the repair cost to show a currency symbol
3. Update the maintenance type to show “Routine” or “Unscheduled” instead of true or false (hint: take a look at apply() and partial evaluation)