CloudCrossing BVBA
Dahlialaan 1
2950 Kapellen
Belgium

FLOW INTRODUCED VARIABLES

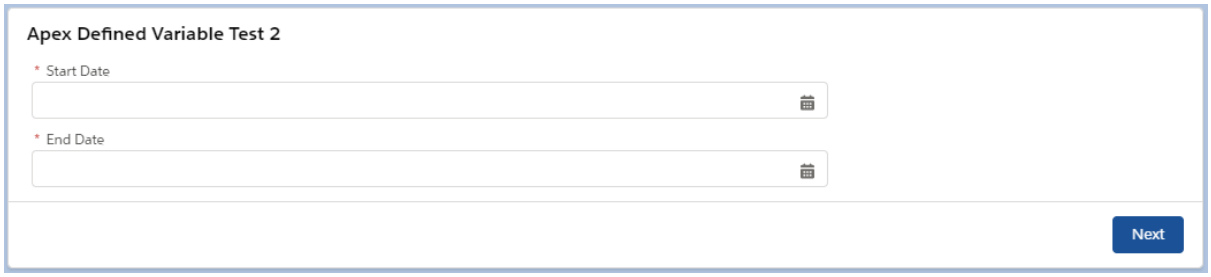
Flow Introduced variables

It can be that the input fields of a Screen Flow would be used as variables in the DataSource queries.

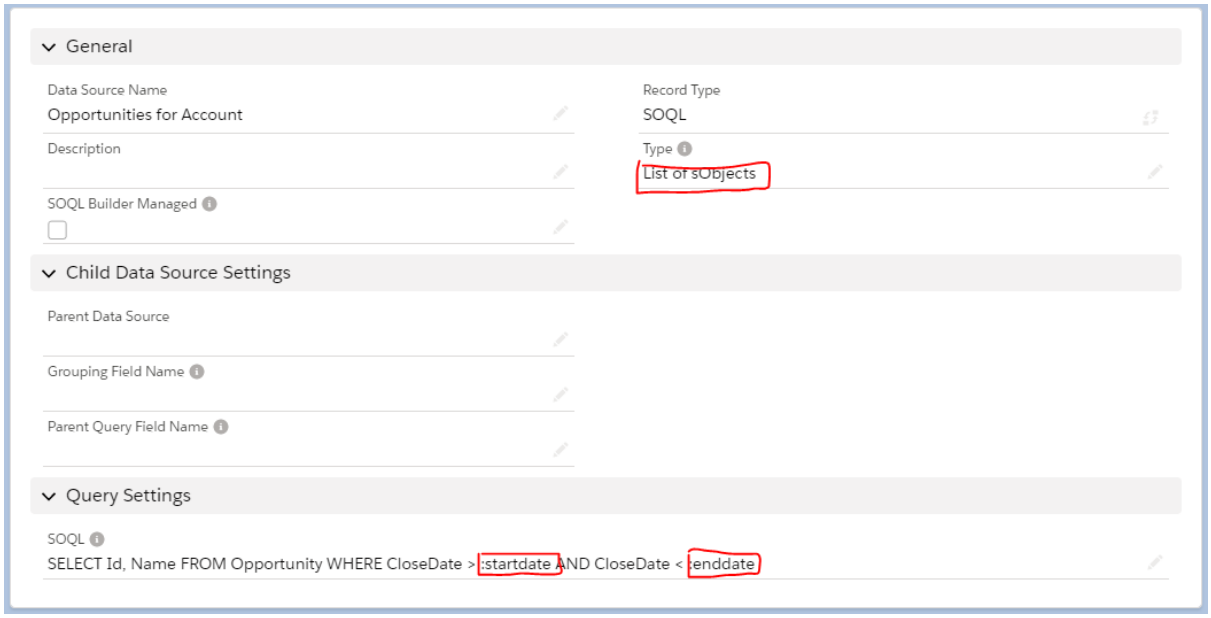
This is possible via Flow Introduced variables.

Let's take a flow as an example.

We want to output all opportunities with a close date in a certain period.



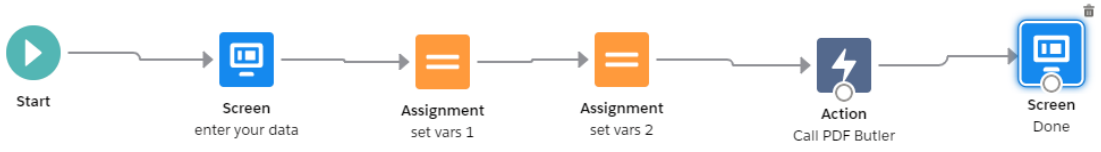
The SOQL we want to run to select the data:



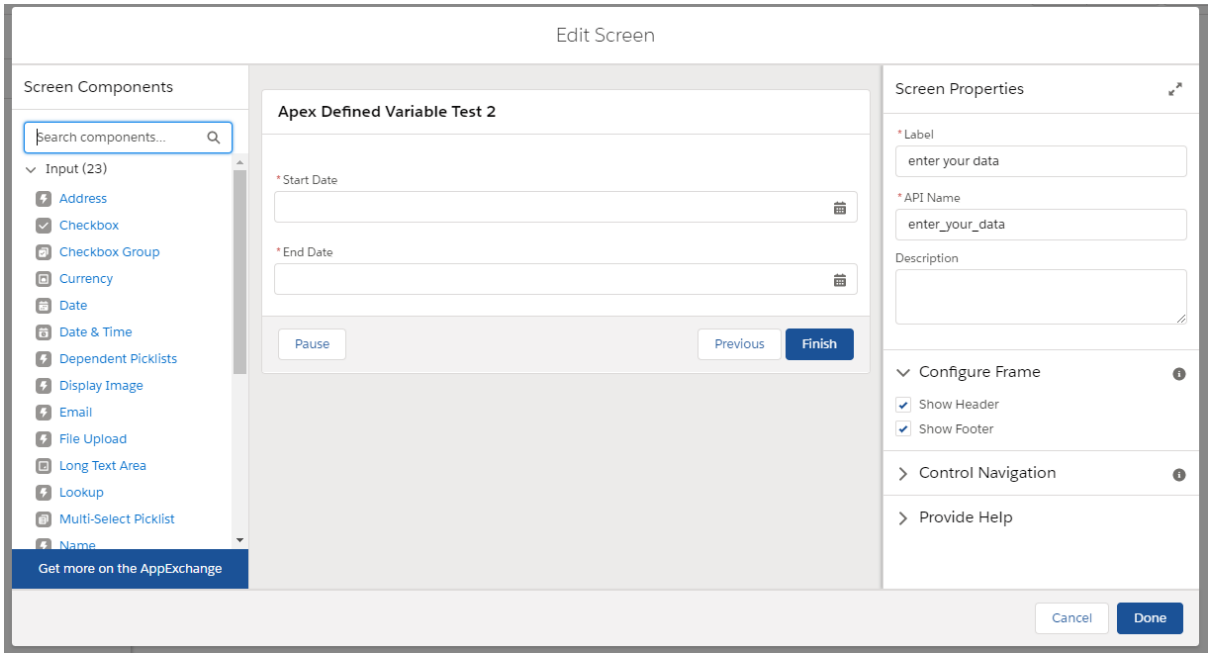
This SOQL takes 2 parameters, the ":startdate" and the ":enddate". These parameters have to be set by the Flow.

Just as the default parameter ":recordId", the parameters must be identified by ":".

The flow:

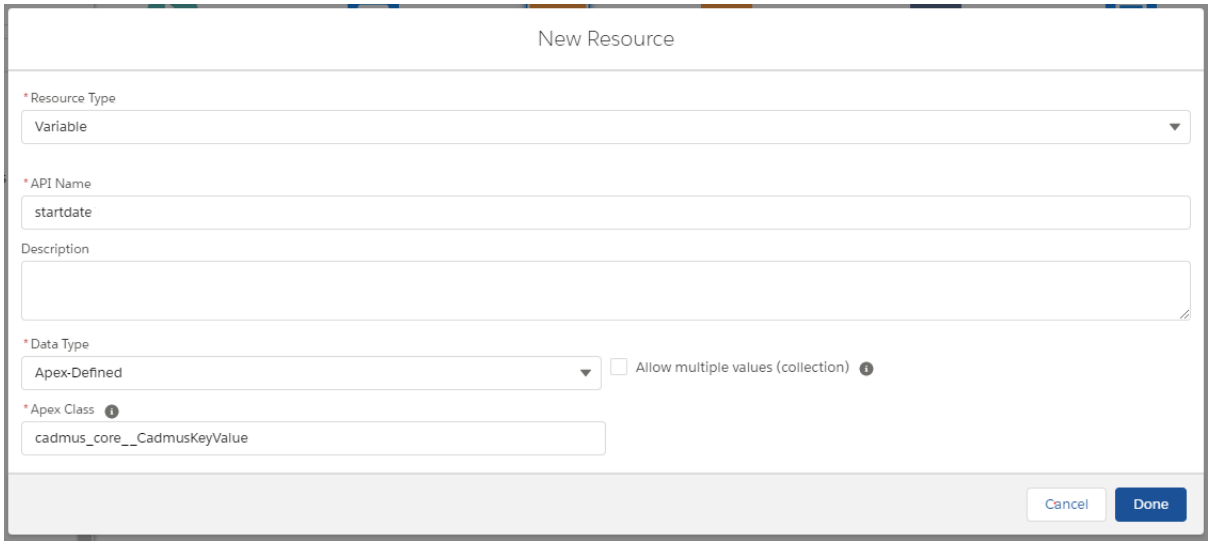


Screen 1 will select the dates:



These are Date fields that allow to select a date, in this case without a time.

Now we want to add these as parameters. So for every parameter, we must create a variable in Flow. The variable must be of type Apex-Defined and set the class cadmus_core__CadmuskeyValue.

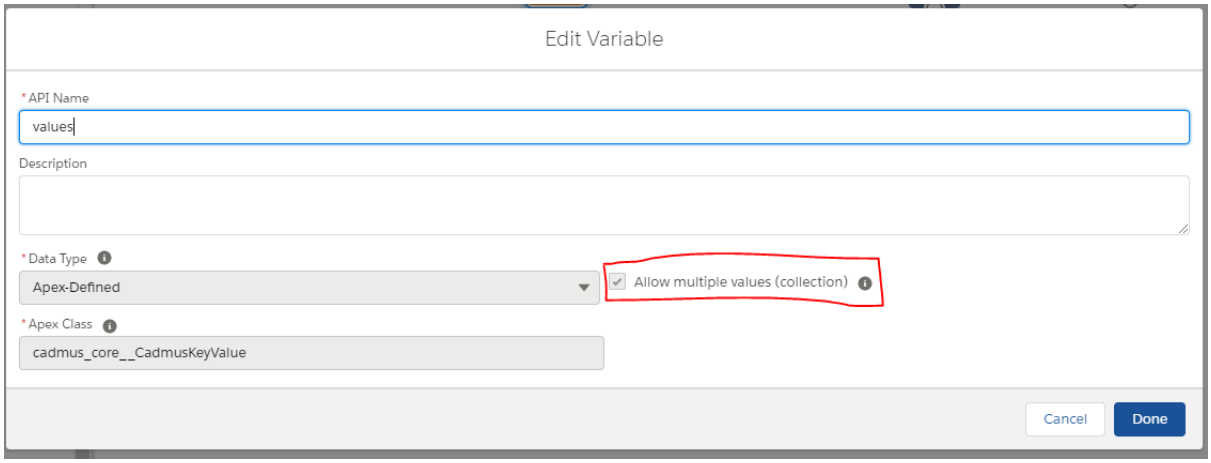


Do the same for the “enddate” parameter.

These parameters have to wrapped in first a collection and second in a parameter variable to pass on to the “Call PDF Butler” component.

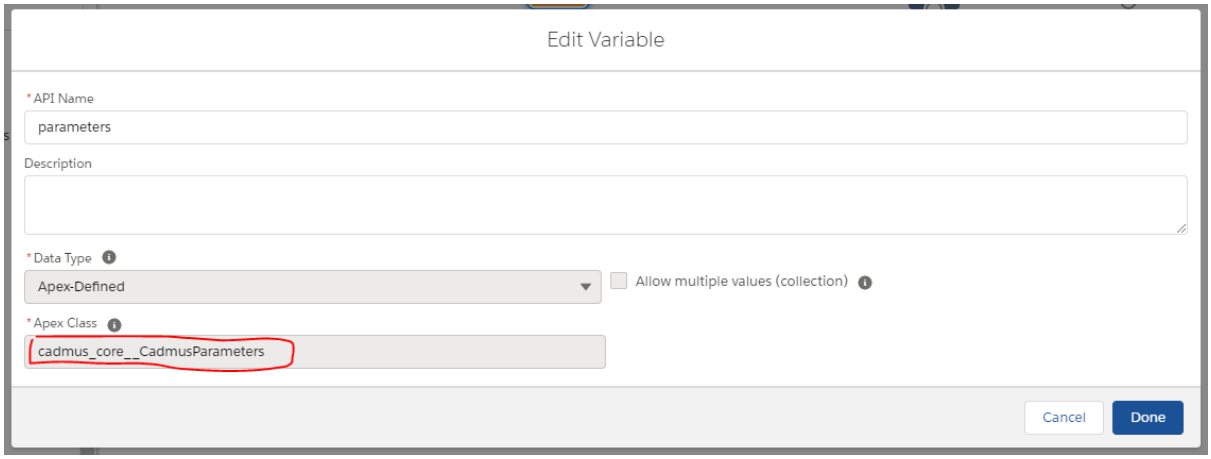
The Collection variable:

See the type, but more important, indicate this is a collection.



Next, the variable that wraps the collection:

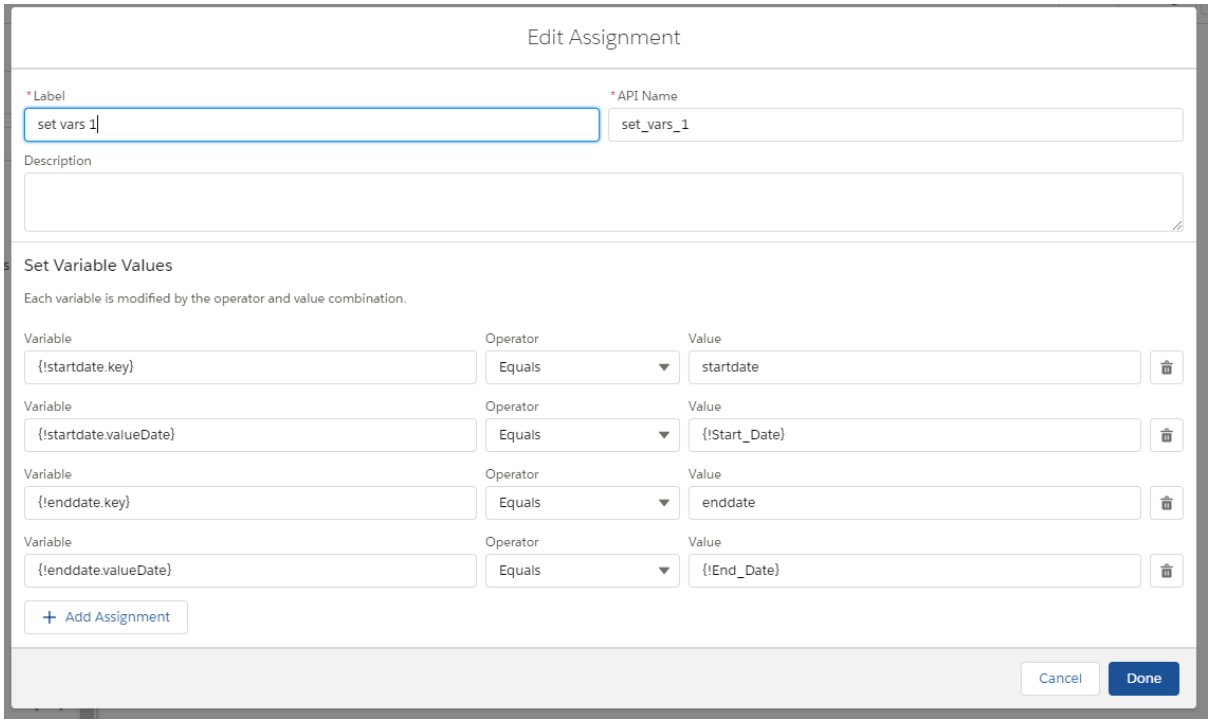
This one has to be of type: “cadmus_core__CadmusParameters”



Assign the values to the variables:

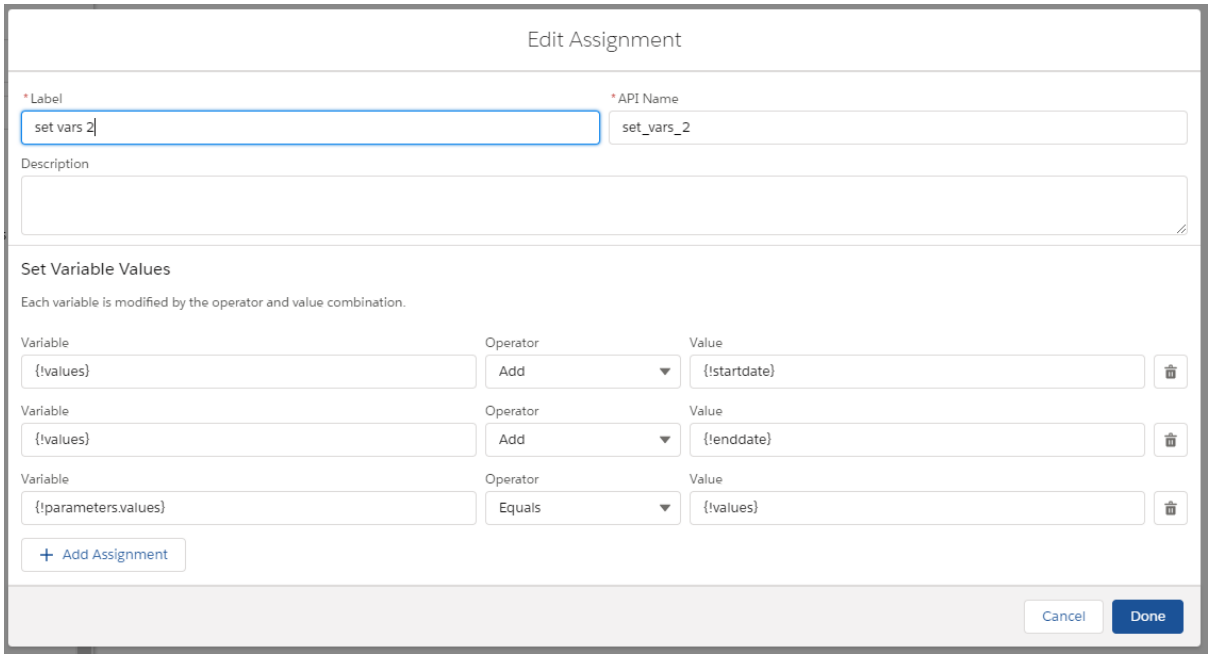
Important to note, the CadmusKeyValue has to be set:

1) key: unique identifier and also maps on the parameter for the SOQL
2) value: this has to be specific for the type of the variable
    a. valueString
    b. valueDouble
    c. valueDate
    d. valueDateTime
    e. valueBoolean



Assign even more variables to fill the wrappers:

- add the variables to the collection
- set the collection to the “values” property



Call PDF Butler

Add an action of type "Call PDF Butler"

Set at least following input values (these are hard coded for demo purposes).

Most important one is when the {!parameters} are set. This is the wrapper variable we set earlier.