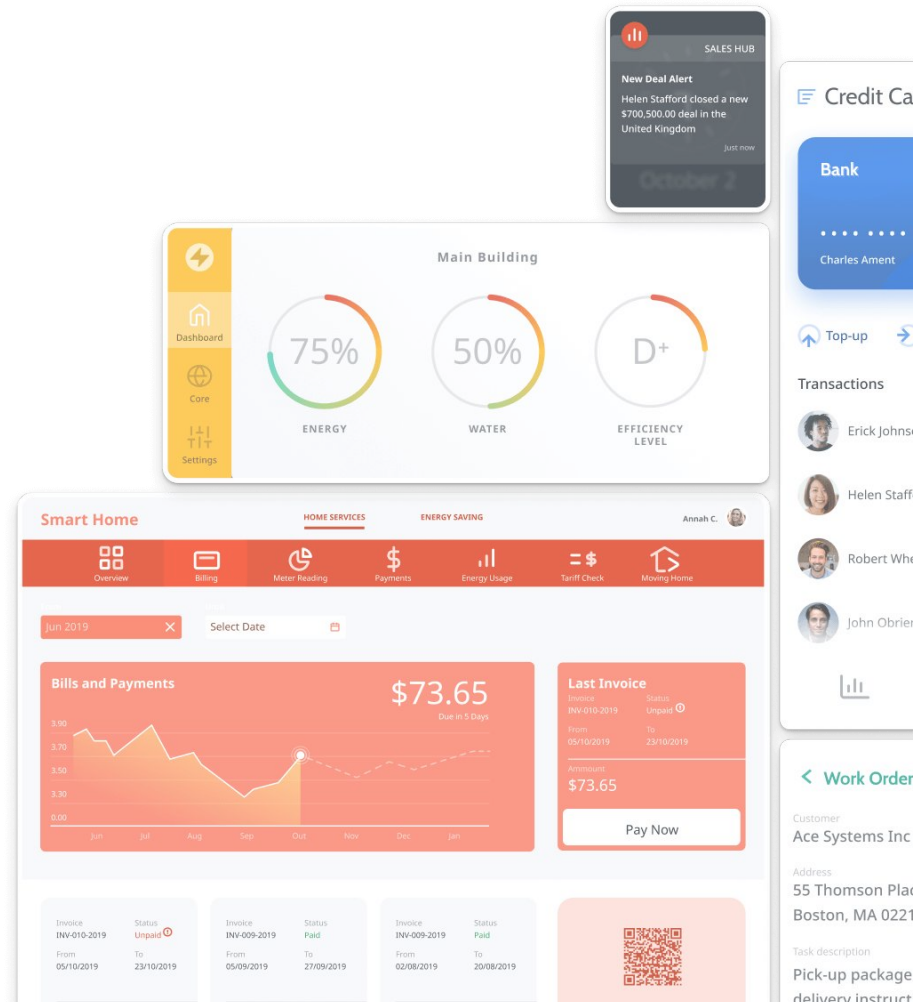


# Logic Flows

Reactive Developer Boot Camp

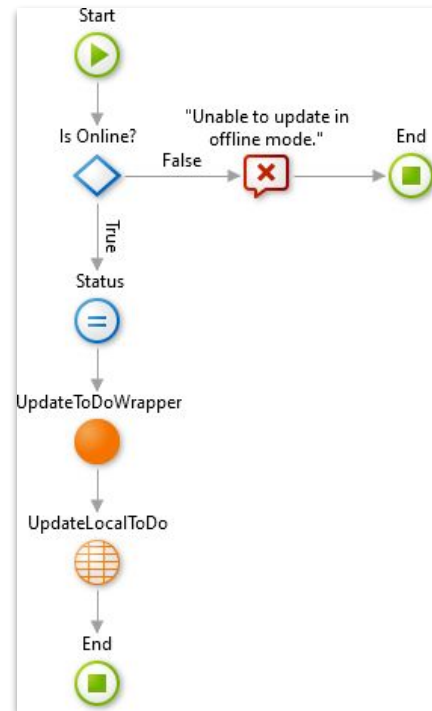


# What you will learn here

- Assign
- Calling Other Actions
- Conditional Branching
- Loops
- Queries

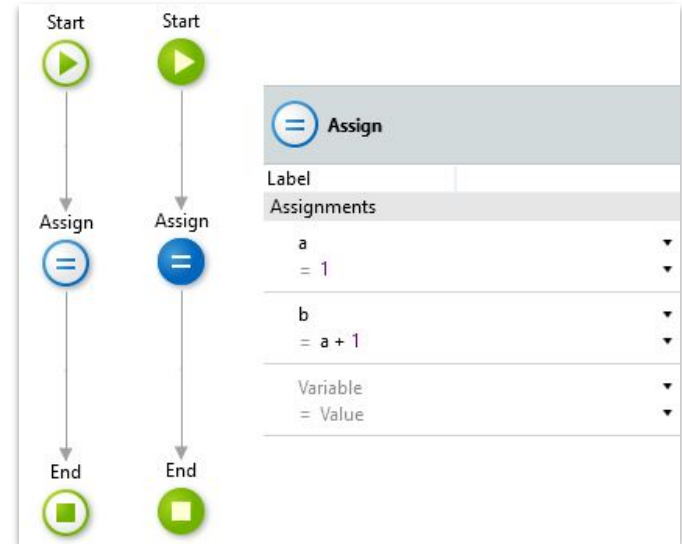
# Action Flows

- An Action Flow is where a piece of logic is defined
- It can only have one Start node
- Every Action Flow can end with multiple nodes
  - End
  - Destination (Screen Actions only)
  - Download (Screen Actions only)



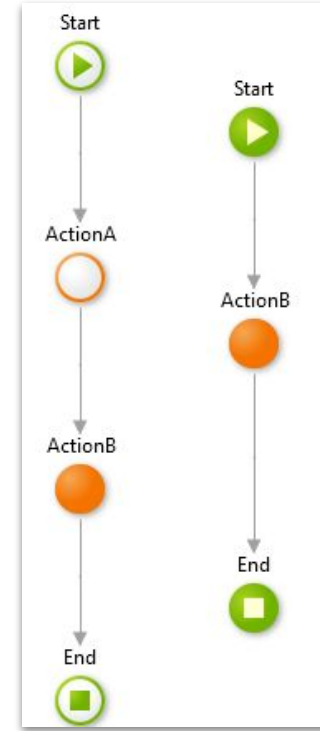
# **Assign**

- Allows setting values to variables (or parameters)
- A single **Assign** can define more than one assignment
  - Values are assigned top to bottom
  - Changes occur immediately
- Service Studio provides standard type-matching values suggestion



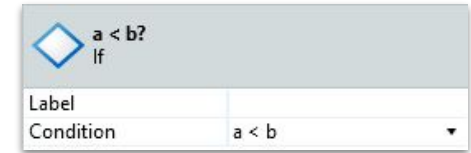
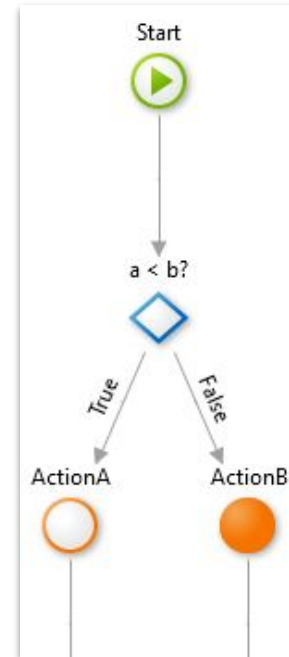
# Calling Other Actions

- A Client Action can call other Client Actions and Server Actions
  - No more than one Server Action call is recommended
- A Server Action can only call other Server Actions



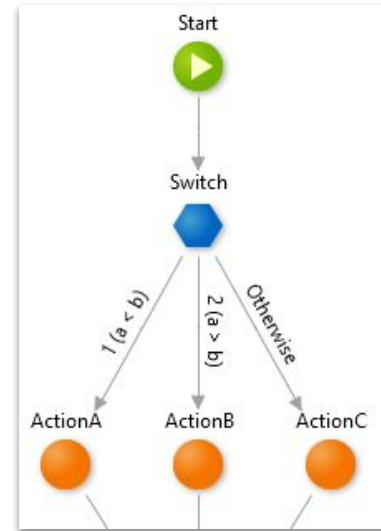


- Creates a conditional branching on an Action flow
- The If condition is evaluated
- Only the corresponding branch is followed depending on the outcome



# Switch

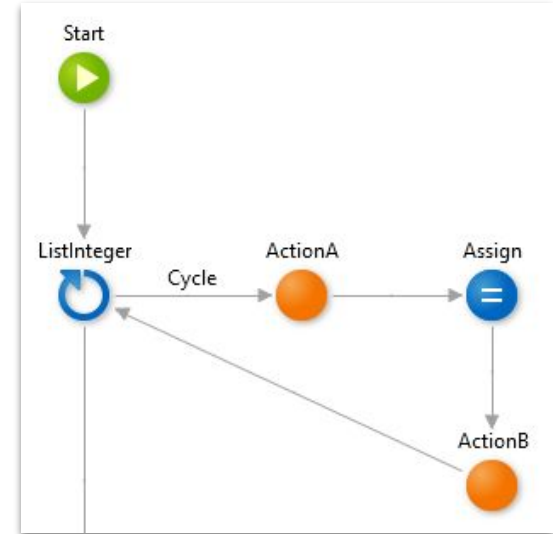
- Creates conditional branching with multiple branches
  - Conditions are evaluated from first to last
- **Only** the first branch that evaluates to True is executed, or the Otherwise branch
- *Otherwise* branch is **mandatory**




Switch	
Label	
Conditions	
Condition 1	a < b
Condition 2	a > b

# For Each

- Allows iterating through a **Record List**
- In the Action flow
  - 1 *Cycle* branch is followed for each record in the List
    - The branch must return to the For Each to continue the loop
  - 2 Branch followed after cycle completes
- *RecordList.Current* gets the record being iterated within the loop

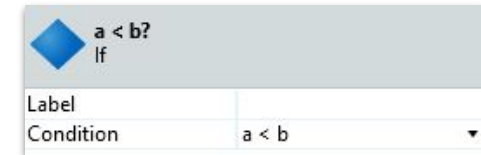
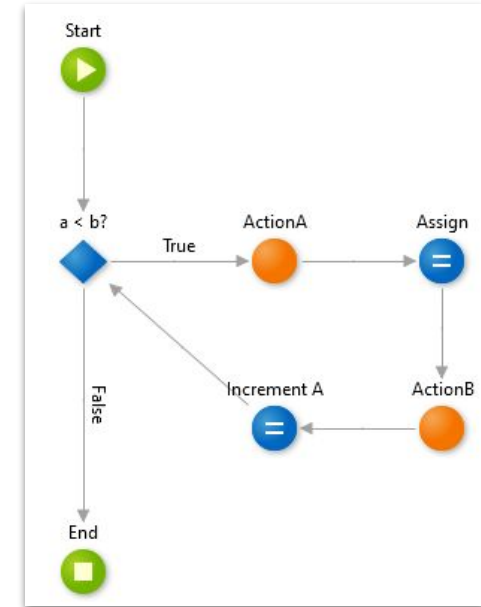


 <b>ListInteger</b> For Each	
Label	
Record List	ListInteger ▼
Start Index	▼
Maximum Iterations	▼



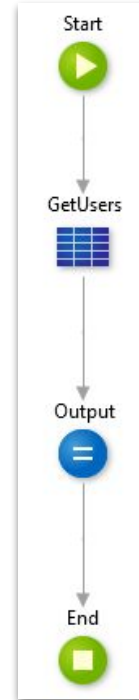
# Ad-hoc Loops





- Use an **If** to evaluate a loop condition
  - Follow a cyclic branch when condition is true
  - Exit loop when condition is false
- In the cyclic branch
  - The branch must return to the If to continue the loop
    - But it can still create alternative branches
  - Be careful with **infinite** loops!



# Aggregates

- A logic flow can also have Aggregates
  - Can be used directly only in Server Actions
  - Defined just like a Screen Aggregate
- Aggregates are executed synchronously and following the order in the Action flow



 GetUsers Aggregate		
Name	GetUsers	...
Description		...
Timeout	(Module Default Timeout)	
Cache in Minutes		
Max. Records		▼
Sources		
 User		...
Filters		
 User.Is_Active = True		...
Sorting		
 User.Name (ASC)		...

# Questions?



Thank You!

