outsystems

Form Validations

Reactive Developer Boot Camp

What you will learn here

- Built-in Validations
- Custom Validations
- Validation Messages

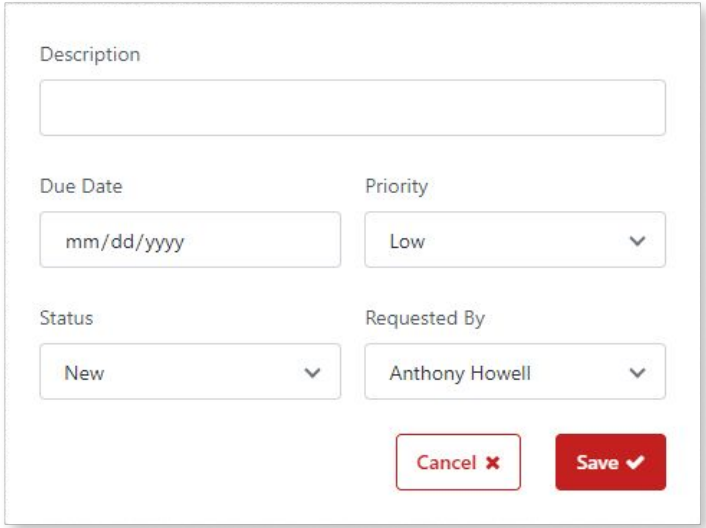
Validating User Inputs

- Applications should limit user mistakes

  → **But they are inevitable**

- Input fields may...
  - Have different data types
  - Depend on other user input values
  - Require business-specific validations

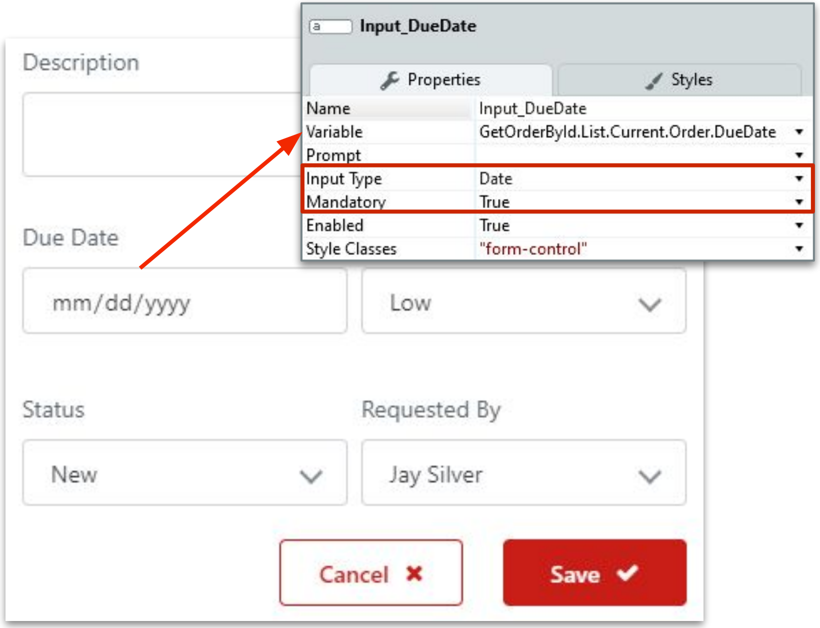
- OutSystems provides mechanisms to help implement input validation



Description	
Due Date: mm/dd/yyyy	Priority: Low
Status: New	Requested By: Anthony Howell

Built-in Validations

- Built-in validations for Input Widgets are performed automatically
    - **Mandatory fields** must be filled
    - Input values must **comply with the data types** of the Variables bound to the widgets

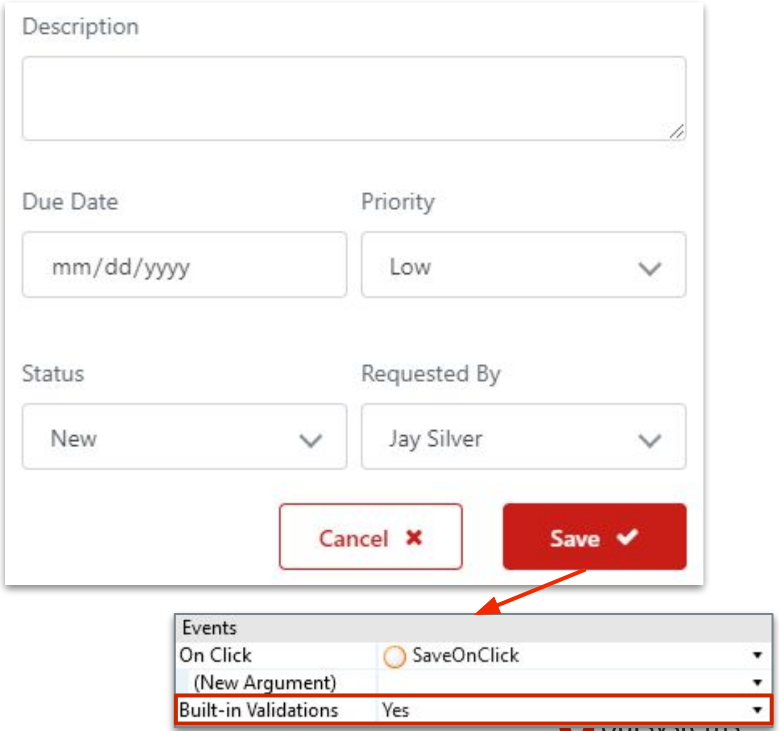


Input_DueDate

Name	Input_DueDate
Variable	GetOrderById.List.Current.Order.DueDate
Prompt	
Input Type	Date
Mandatory	True
Enabled	True
Style Classes	"form-control"

Built-in Validations

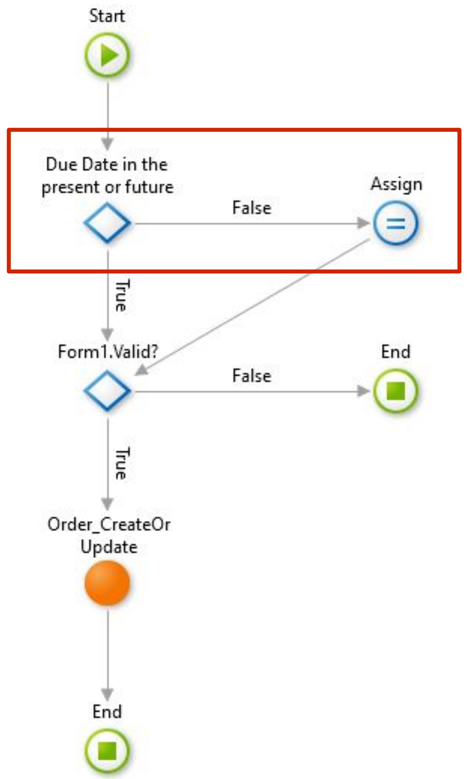
- Built-in validations for Input Widgets are performed automatically
  - **Mandatory fields** must be filled
  - Input values must **comply with the data types** of the Variables bound to the widgets
- Buttons / Links inside a Form have a Built-in Validations Property
  - Set to **Yes** to perform built-in validations



Events	
On Click	SaveOnClick
(New Argument)	
Built-in Validations	Yes

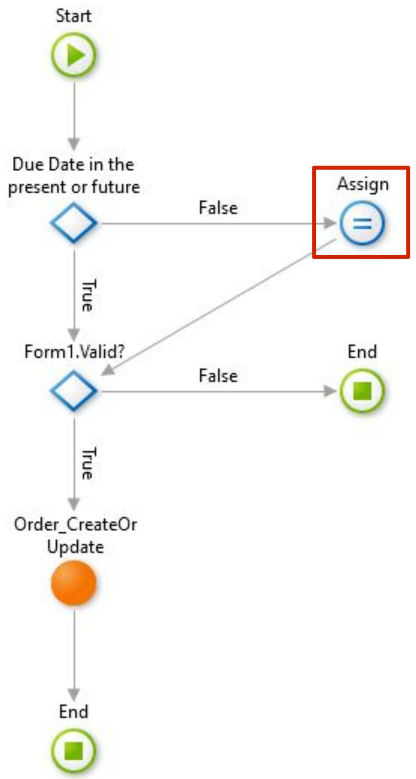
Custom Validations

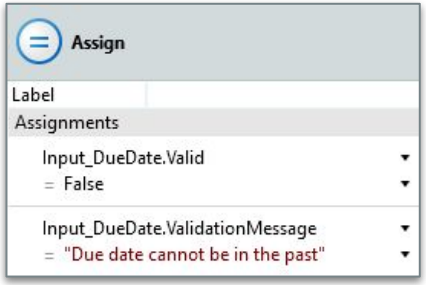
- Inputs can have custom validations
  - Must be performed in the Action flow
  - For invalid Form Input fields
    - Set the **Valid** property to False
    - Set the **ValidationMessage**



Custom Validations

- Inputs can have custom validations
  - Must be performed in the Action flow
  - For invalid Form Input fields
    - Set the **Valid** property to False
    - Set the **ValidationMessage**

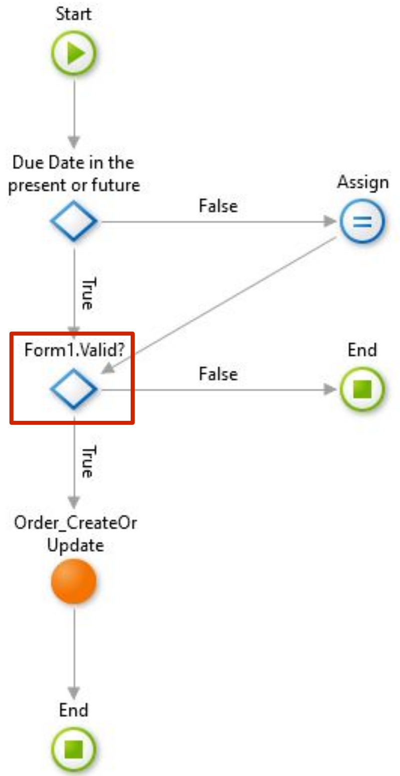


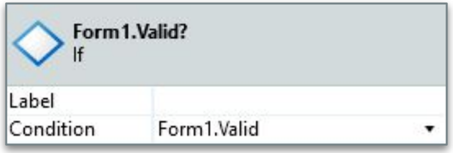

Assign

Label	
Assignments	
Input_DueDate.Valid	= False
Input_DueDate.ValidationMessage	= "Due date cannot be in the past"

Custom Validations

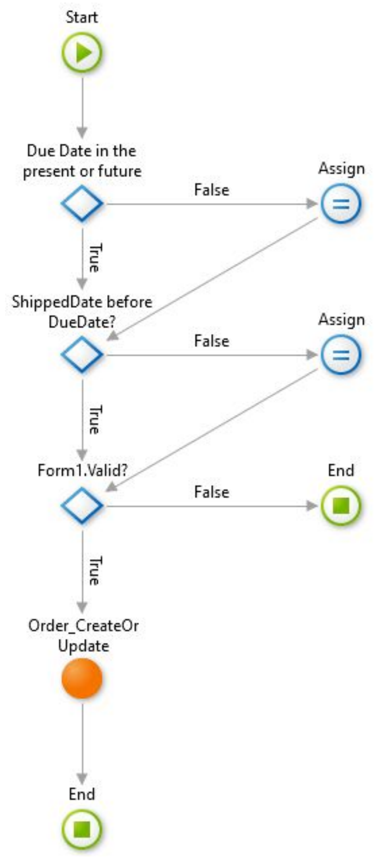
- Inputs can have custom validations
  - Must be performed in the Action flow
  - For invalid Form Input fields
    - Set the **Valid** property to False
    - Set the **ValidationMessage**
- The **Form.Valid** property should be checked after the last custom validation
  - If one Input is not Valid, the Form is automatically not Valid
  - The *Form.Valid* cannot be explicitly changed



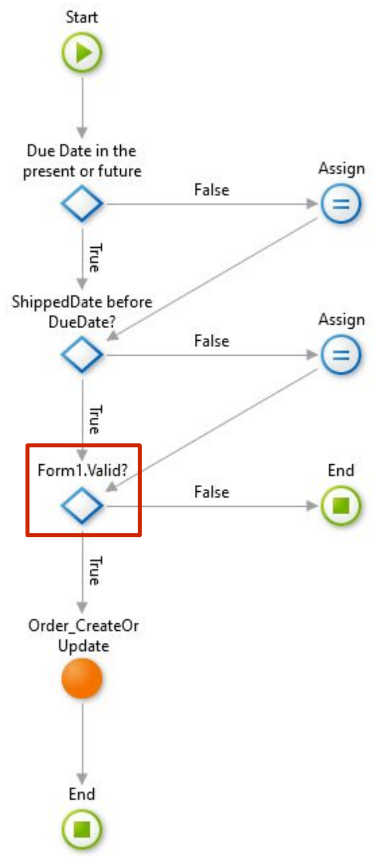

Form1.Valid?
If

Label	
Condition	Form1.Valid

Multiple Validations

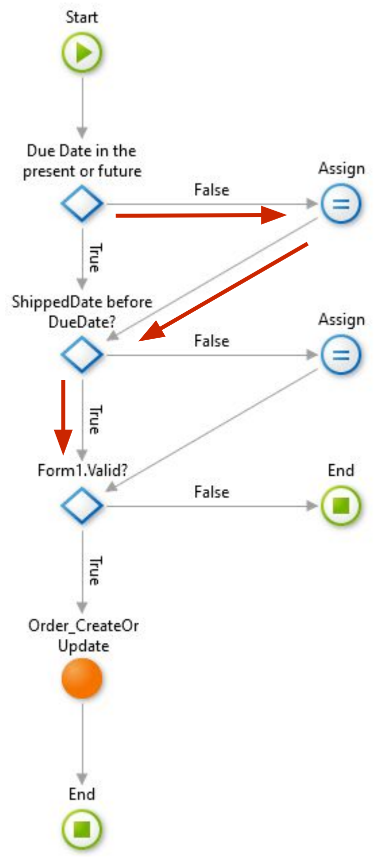


Multiple Validations



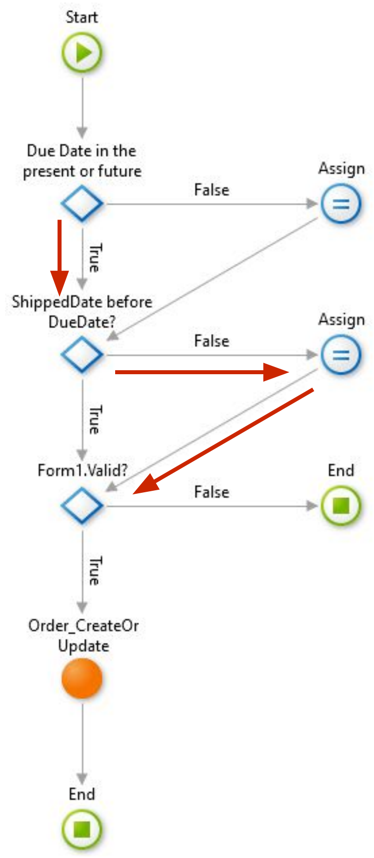
Multiple Validations

- First validation fails
- Second one passes



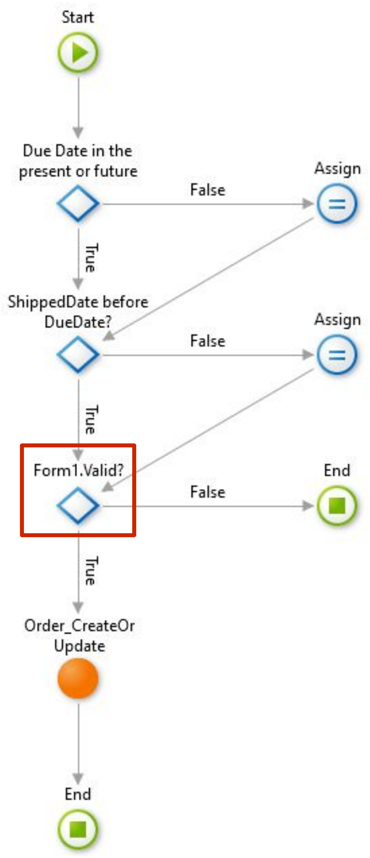
Multiple Validations

- First validation passes
- Second one fails



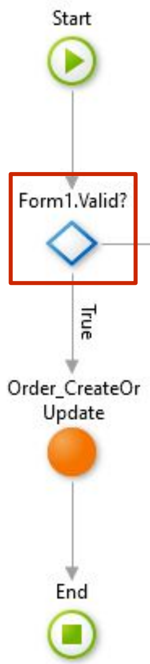
Multiple Validations

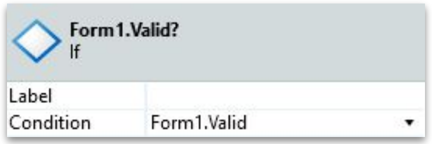
- If the Form is not valid the flow ends
- If it is valid, data can be safely used in logic



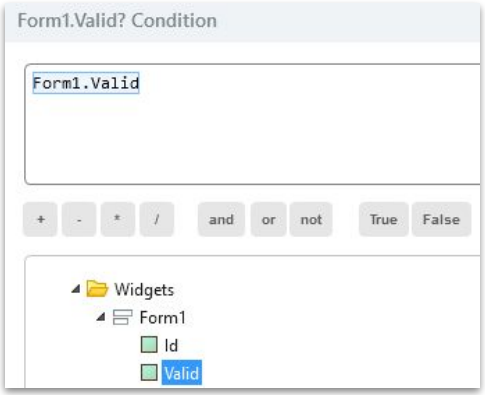
Checking Built-in Validations

- The *Form.Valid* property should be checked even if there are no custom validations
    - To avoid storing / manipulating invalid data
- OutSystems automatically performs the validations
    - But it is up to the developer to leverage that in the logic





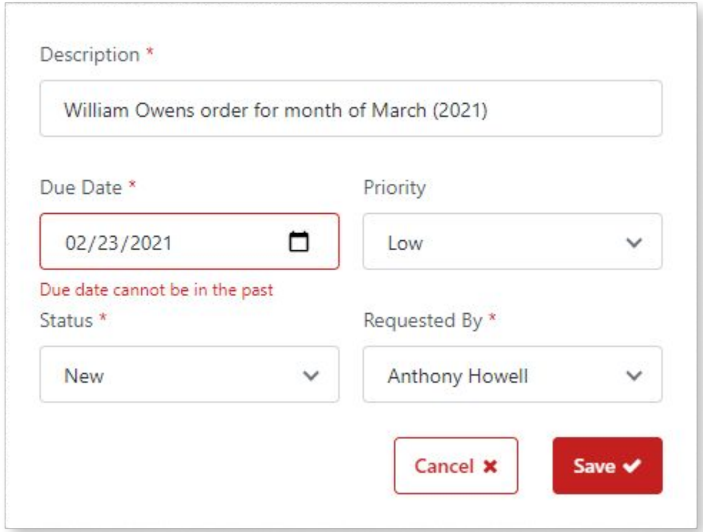
Label	
Condition	Form1.Valid



```
Form1.Valid
```

Validation Messages

- The Valid property of the Inputs are checked when rendering the Screen
- If Valid property is set to True:
  - Displays the regular widget
- If Valid property is set to False:
  - Displays the regular widget
  - Applies the specific styling
    - e.g. a red border



Description *

William Owens order for month of March (2021)

Due Date *

02/23/2021

Due date cannot be in the past

Priority

Low

Status *

New

Requested By *

Anthony Howell

Questions?

outsystems

Thank You!