Flow Variables and Form Validation

Table of Contents

- 1 Objective
- 2 Do the following task items
- 2.1 Using flow instance variables
  - 2.1.1 Configure the image URL variables
    - 2.1.1.1 Define flow variables directly
      - 2.1.1.1.1 Using predefined flow variables in a flow
  - 2.1.2 Change HTTP message nodes to use variables
    - 2.1.2.1 Test that images display
  - 2.1.3 Add variable for user first name
  - 2.1.4 Use the variable in Collect Age
    - 2.1.4.1 Test your change that Collect Age page will show first name
- 2.2 Form validation
  - 2.2.1 Validate entry with property rules
    - 2.2.1.1 Test validation changes for first name
  - 2.2.2 Validate entry with functions and use error message component
    - 2.2.2.1 The template for collect age needs component to display the message
    - 2.2.2.2 Test the age validation check

1 Objective

In this exercise you will cover two concepts to further expand the functionality of your existing flow.

- Using flow variables to define things that may be referenced in multiple places in a flow or be easier to manage if centralized in one place for changes.
- Improve the interaction with the user by providing better form validation in the forms where you accept input from the user.

Tip

All the number instruction steps in each section must be followed to successfully complete this exercise. Any bullet points or images are for example only unless called out in a numbered step.

2 Do the following task items

The two major topics of this flow will be covered in two sections using variables and improving form validation inputs.

2.1 Using flow instance variables

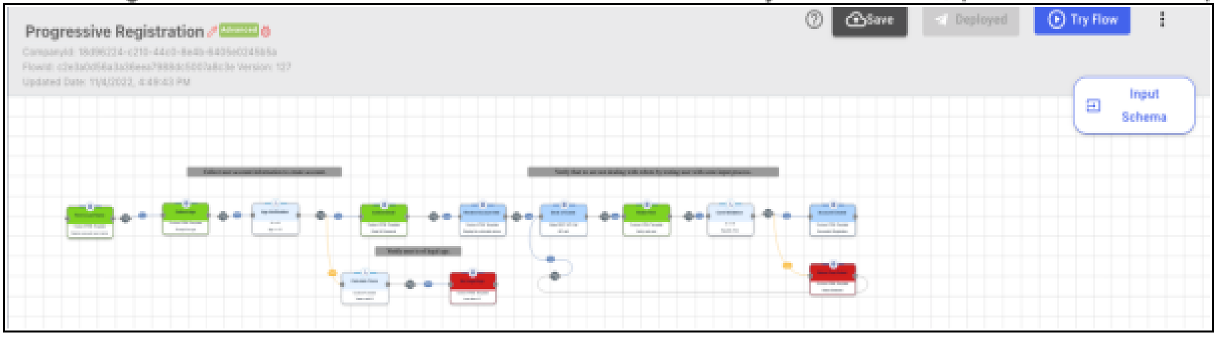
In building out the flow you may want to build out a user object that holds the data as you collect it. This way when you go to actually create the object in some directory such as PingOne. You can fetch the information from one place rather than multiple nodes.

Another potential use case is for data that is used in different nodes and may change. It can be either reused in multiple nodes or just simply used once but you don't want to remember where.

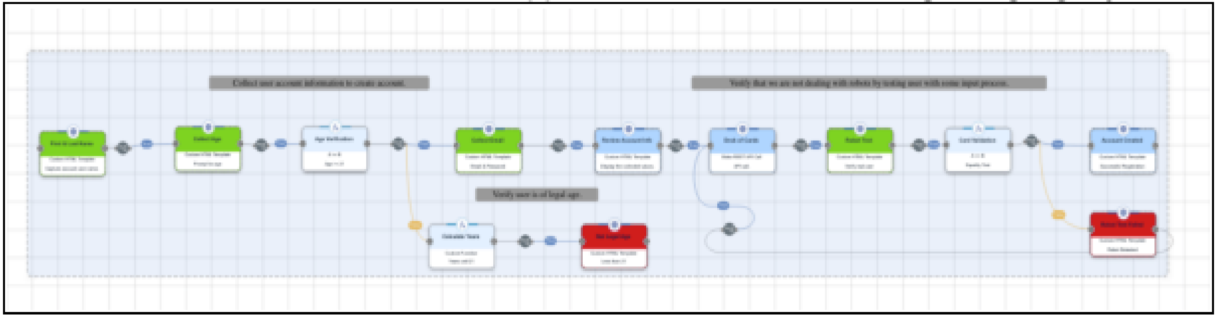
These are good candidates for a variables that can be defined in one place and used anywhere in the flow. Typically you want these variables to be available anywhere in the flow, so it is best to define it as the first node in the flow.

1 **Open** your **Progressive Registration** flow.
   - You can optionally clone your flow to back it up before you start. Don't forget to rename the clone as a backup.

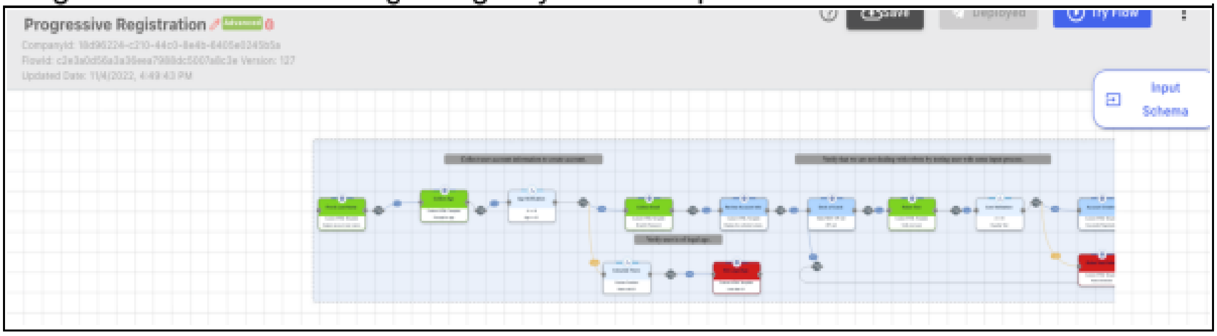
2 You are going to add your new node at the start of your flow. Use the **-** next to the **Reset** at the bottom right of the canvas to reduce the flow to all fit in your browser, plus some extra space


Progressive Registration
Advanced
Save
Deployed
Try Flow
Input Schema
Collect user account information to create account.
Verify that we are not dealing with robots by testing user with some input process.
Verify user is of legal age.

3 Click and hold the **left-mouse** in the upper left corner and then drag to highlight your entire flow


Collect user account information to create account.
Verify that we are not dealing with robots by testing user with some input process.
Verify user is of legal age.

4 Drag the entire flow to the right to give you some space

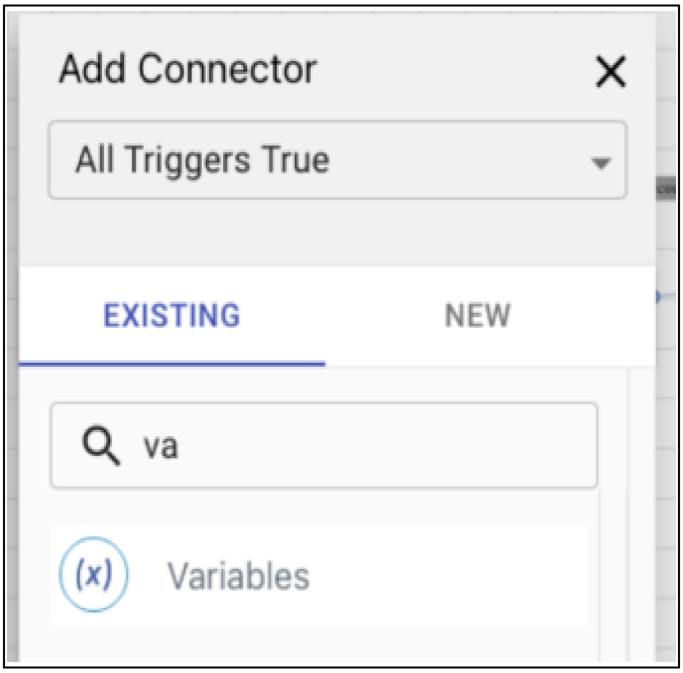


5 Click anywhere on the canvas to un-highlight the selected flow. And don't hit the reset button.

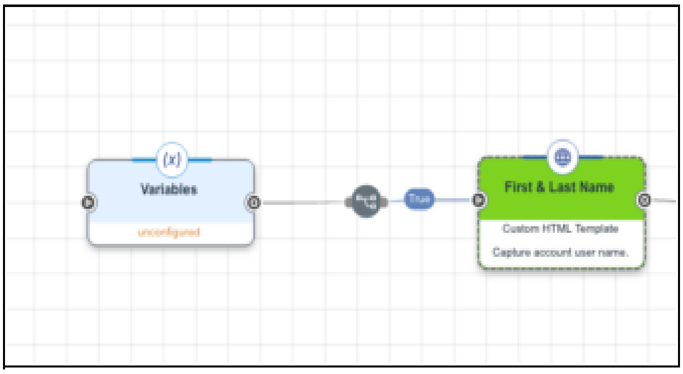
6 From the left of the first node on the left (**First & Last Name**) click and drag a path line.

- This may take a couple of attempts to get a path line and could even end up moving the node it self depending on your mouse pointer precision skills.
- Alternatively you can use the **+ icon in the bottom left to add the node on the canvas and then connect to the rest of the flow at the start.**

7 Search for **Variables** connector and select it.



8 You can click **Reset** button in the bottom left to reset the flow to a more reasonable readable size.



9 Good time to save your flow.

10 Click the **Variables** node you just added at the start of your flow.

- In a lot of flows a variables node is a handy starting point, so not a bad idea to start with one.
- You can always delete it you don't use it in your flow when you are done.

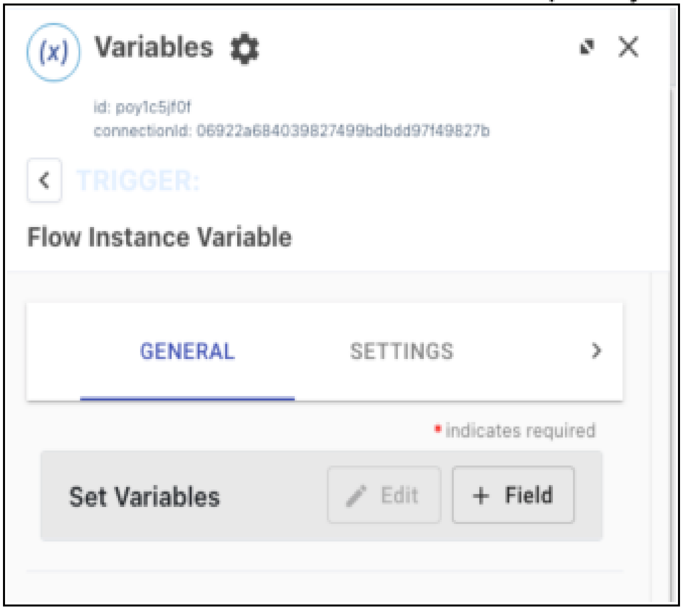
11 Continue with the next section.

2.1.1 Configure the image URL variables

In this section you will define three images that are referenced in the HTML templates you added to the HTTP nodes in the previous exercise.

1 Click **Variables** node you added in the previous section.

2 Click **Flow Instance Variable** as the capability.



id: poy1c5jf0f
connectionId: 06922a684039827499bdbdd97f49827b

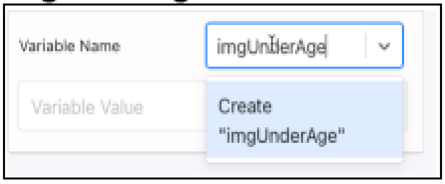
Flow Instance Variable

Set Variables

3 Click **+ Field** button to the right of Set Variables, this will create a new line with Variable Name and next to it a field with *Select or Type* in it.

- ○ Variables are stored globally in DaVinci and you can reference these in different flows and set them for the flow.
- ○ At this time you have none in your deployment and will be creating new ones for this flow.

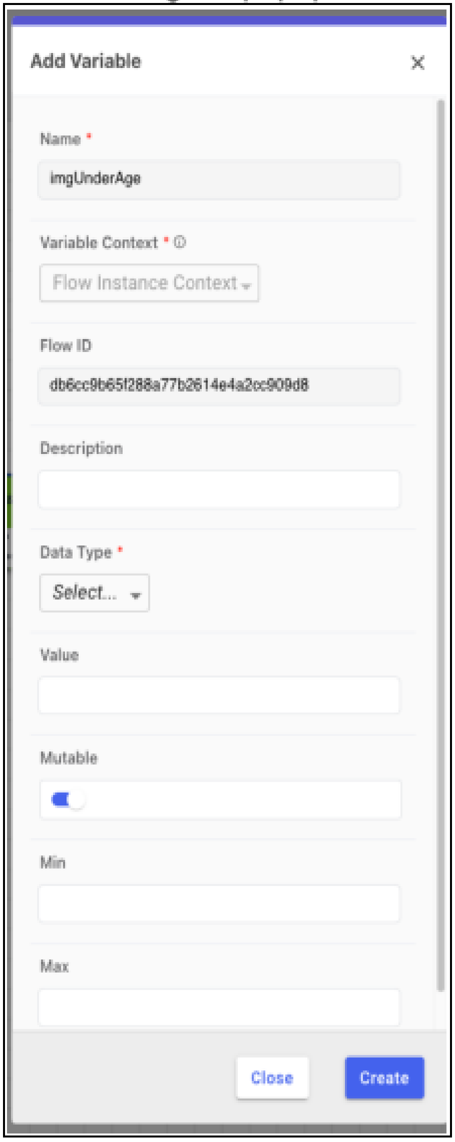
4 In the **Variable Name** field copy and paste the following:

- ○ **imgUnderAge**
- ○ 

Variable Name: imgUnderAge

5 Press **Return** to create the new variable.

- A new dialog will popup to the left.



Add Variable

Name * imgUnderAge

Variable Context * Flow Instance Context

Flow ID db6cc9b65f288a77b2614e4a2cc909d8

Description

Data Type * Select...

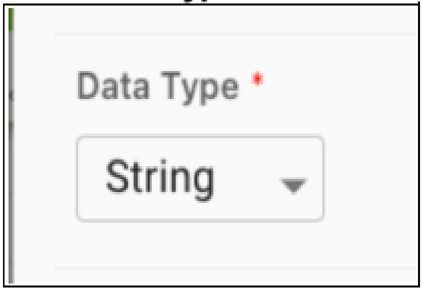
Value

Mutable

Min

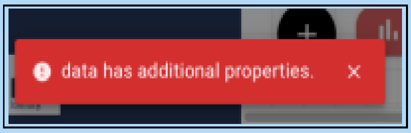
Max

5.1 In the **Data Type** field select **String** from the dropdown list.



Data Type * String

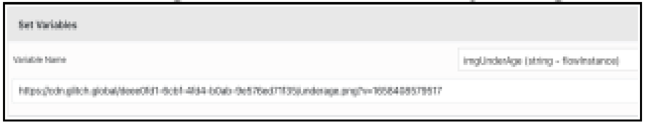
6 Click **Create** button to save and close the dialog.

> **Note**
>
> If you run into an error when creating the variable about additional properties. Then please can continue to the next sub-section and create the variables directly.
>
> 
>
> data has additional properties. ×
>
> If no error then continue in this section of the exercise. Just use the next two sub-sections as references for future consideration in development of flows.

7 In the **Variable Value** field copy and paste the following text, remember no spaces (or line feeds) before or after when you paste it.

```
https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/underage.png?v=1658408579517
```

- The following view shows the entry using the expanded view of the dialog as an example.



Set Variables

Variable Name: imgUnderAge (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/underage.png?v=1658408579517

8 You need to define two more variables using the above steps with the following information:

8.1 New Field where:

8.1.1 **Variable Name** is: **imgRobotDetected**

8.1.2 In the **Data Type** field select **String** from the dropdown list.

8.1.3 **Variable Value** field copy and paste the following text, remember no spaces (or line feeds) before or after when you paste it.

```
https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/robot-detected.png?v=1658408579586
```

8.2 New Field where:

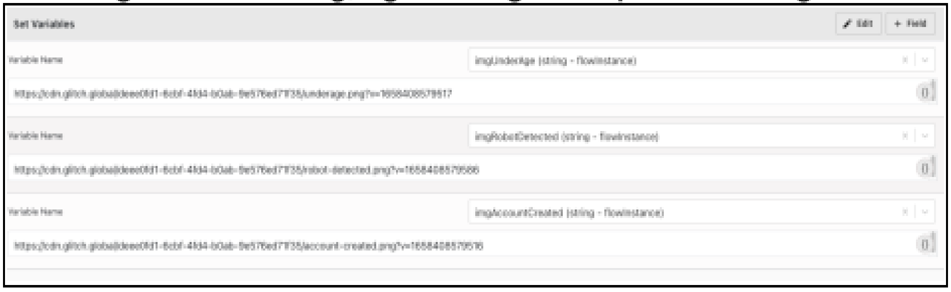
8.2.1 **Variable Name** is: **imgAccountCreated**

8.2.2 In the **Data Type** field select **String** from the dropdown list.

8.2.3 **Variable Value** field copy and paste the following text, remember no spaces (or line feeds) before or after when you paste it.

```
https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/account-created.png?v=1658408579516
```

9 Resulting in the following, again using the expanded dialog view:



Set Variables

Variable Name: imgUnderAge (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/underage.png?v=1658408579517

Variable Name: imgRobotDetected (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/robot-detected.png?v=1658408579586

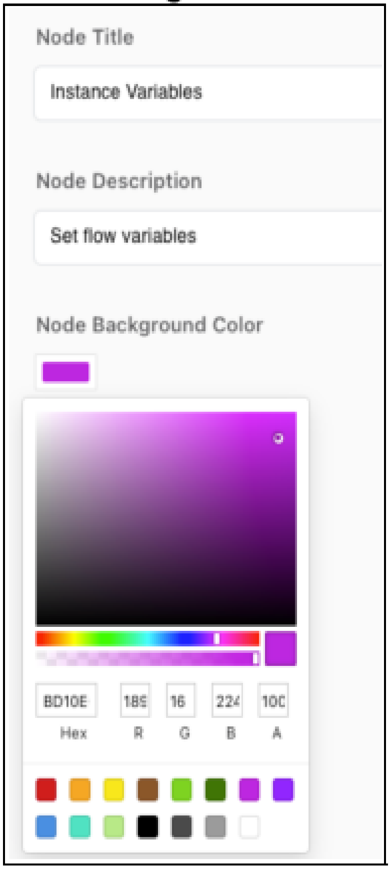
Variable Name: imgAccountCreated (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/account-created.png?v=1658408579516

10 Click **Apply** button to save your changes.

11 In the **Settings** tab enter the following:

- 11.1 **Node Title** is
  - **Instance Variables**
- 11.2 **Node Description** is
  - **Set flow variables**
- 11.3 **Node Background Color** select color of lighter **purple** or **magenta**



Node Title: Instance Variables

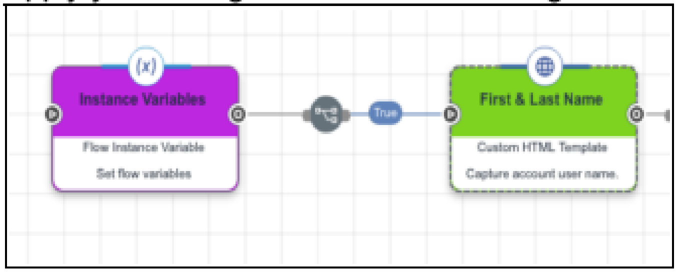
Node Description: Set flow variables

Node Background Color

Hex: BD10E R: 189 G: 16 B: 224 A: 100

  - You can select another color if you prefer the main idea is to use coloring coding to help identify the type of work your nodes are doing.
  - This color coding should be something that is agreed on by everyone in your organization that is doing DaVinci flows.

12 Apply your changes and close the dialog.



13 Click **Save** button to save your flow changes, if needed.

14 Skip the next two sub-sections and continue with the section titled ***Change HTTP message nodes to use variables*** at same level as this one.

- The next two sections describe the steps for declaring variables using the Variables left menu section in the console instead of in a flow.
- This is an alternative approach to defining variables, which you have already completed in this section.

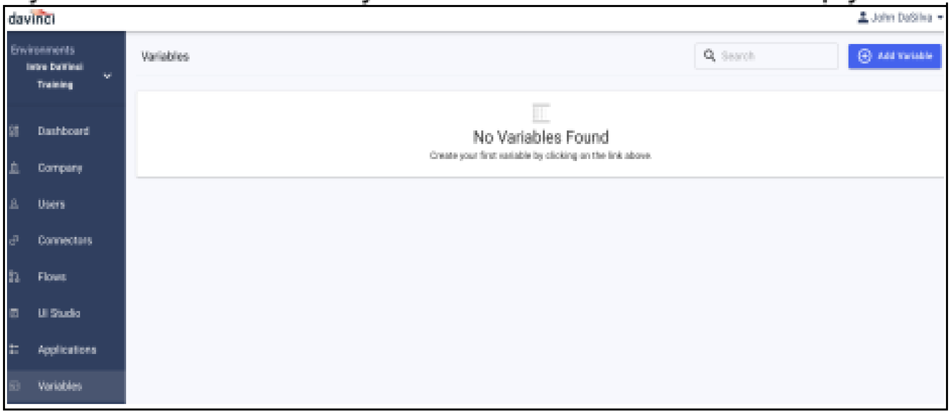
2.1.1.1 Define flow variables directly

Alternatively to defining variables within a flow you can define them directly in the Variables section of the DaVinci admin console. This can be handy especially if you have a lot of variables to define that will be used in multiple flows.

If you have already defined your variables in the above section **you can skip** the steps in this and the next section. The objective of outlining this here is to provide an alternative approach, especially if you know up front what variables you will need or want to standardize across multiple flows.

1 Make sure your flows current state is saved and click the **Variables** menu item in the left menu.

2 If you have not defined any variables this section will be empty.



No Variables Found

Create your first variable by clicking on the link above.

3 Click the **Add Variable** button in the top right corner, a dialog box will open.

4 Configure the variable as follows:

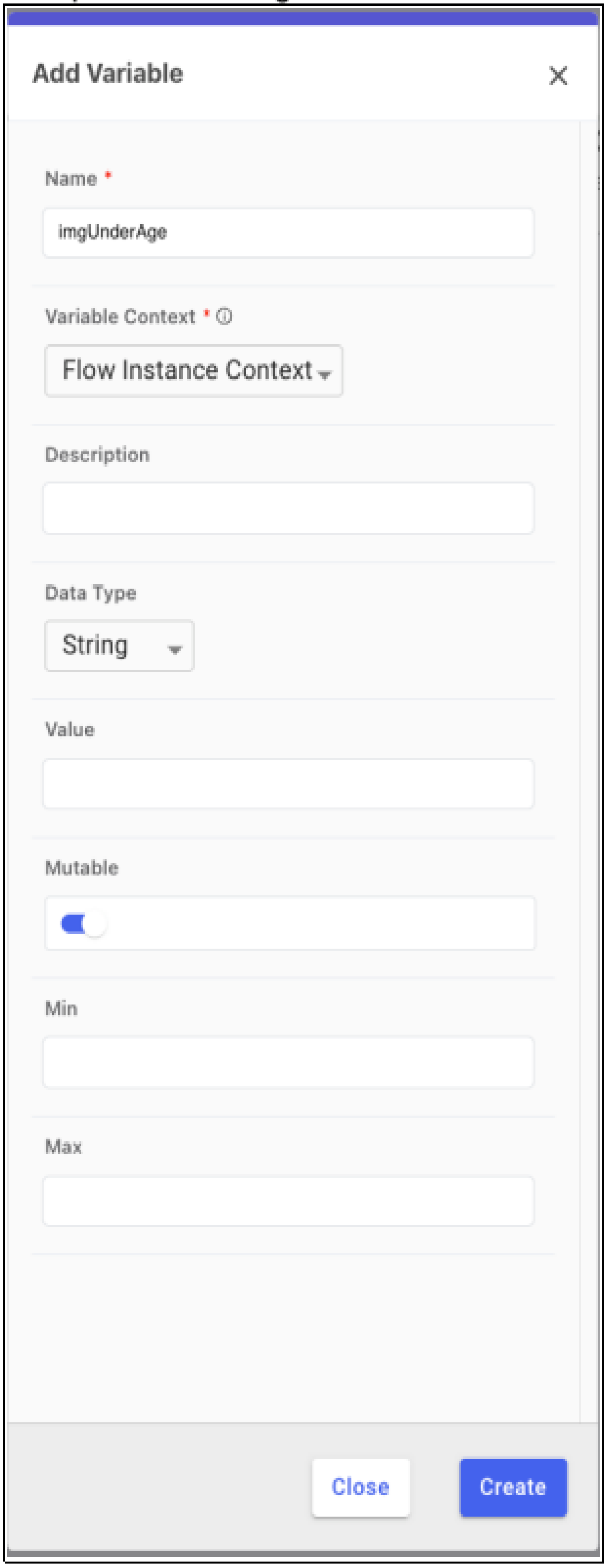
4.1 In the **Name** field copy and paste the following:

- **imgUnderAge**

4.2 In the **Variable Context** dropdown select **Flow Instance Context** value.

4.3 In the **Data Type** field select **String** from the dropdown list.

4.4 Accept the remaining default values.



Add Variable

Name *

imgUnderAge

Variable Context *

Flow Instance Context

Description

Data Type

String

Value

Mutable

Min

Max

- The value information will be provided in the actual flow that will use this variable.

4.5 Click **Create** button to create the variable and close the dialog.

5 Click the **Add Variable** button in the top right corner, a dialog box will open.

6 Configure the variable as follows:

6.1 In the **Name** field copy and paste the following:

- **imgRobotDetected**

6.2 In the **Variable Context** dropdown select **Flow Instance Context** value.

6.3 In the **Data Type** field select **String** from the dropdown list.

6.4 Accept the remaining default values.

6.5 Click **Create** button to create the variable and close the dialog.

7 Click the **Add Variable** button in the top right corner, a dialog box will open.

8 Configure the variable as follows:

8.1 In the **Name** field copy and paste the following:

- **imgAccountCreated**

8.2 In the **Variable Context** dropdown select **Flow Instance Context** value.

8.3 In the **Data Type** field select **String** from the dropdown list.

8.4 Accept the remaining default values.

8.5 Click **Create** button to create the variable and close the dialog.

9 Click the **Add Variable** button in the top right corner, a dialog box will open.

10 Configure the variable as follows:

10.1 In the **Name** field copy and paste the following:

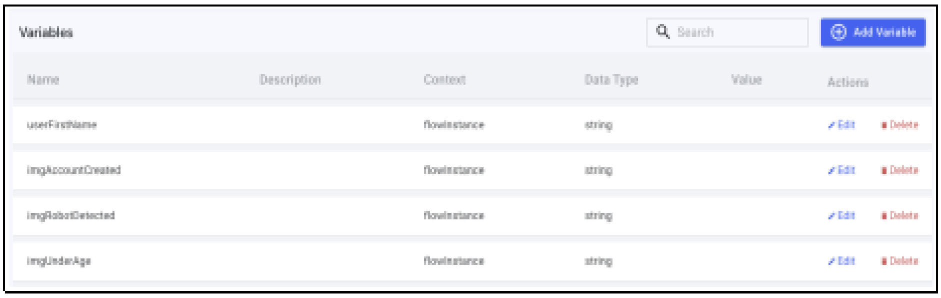
- **userFirstName**

10.2 In the **Variable Context** dropdown select **Flow Instance Context** value.

10.3 Accept the remaining default values.

10.4 Click **Create** button to create the variable and close the dialog.

11 You will have four variables defined.



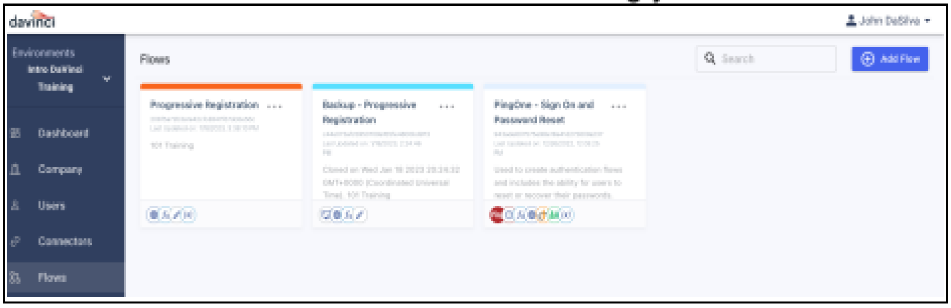
Name	Description	Context	Data Type	Value	Actions	
userFirstName		flowInstance	string		Edit	Delete
imgAccountCreated		flowInstance	string		Edit	Delete
imgRobotDetected		flowInstance	string		Edit	Delete
imgUnderAge		flowInstance	string		Edit	Delete

12 Continue with the next section.

2.1.1.1.1 Using predefined flow variables in a flow

If you have predefined your flow variables then you will reference them when defining a Variables node in the flow rather than creating them.

1 In the left menu click **Flows** to return to editing your flow.



Progressive Registration

101 Training

Backup - Progressive Registration

Cloned on Wed Jan 18 2023 20:24:32 GMT+0000 (Coordinated Universal Time). 101 Training

PingOne - Sign On and Password Reset

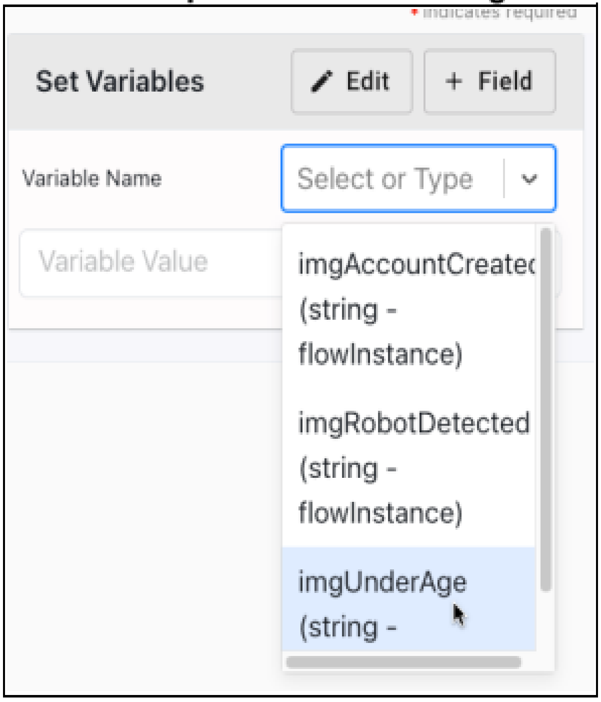
Used to create authentication flows and includes the ability for users to reset or recover their passwords.

2 Click on the **Progressive Registration** flow to edit it.

3 Click on the **Variables** node at the start of your flow to edit it.

4 Click **+ Field** button to the right of Set Variables, this will create a new line with Variable Name configure it as follows:

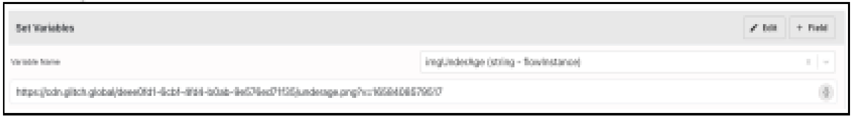
4.1 From the **dropdown** field select **imgUnderAge** from the list to configure for this flow:



4.2 In the **Variable Value** field copy and paste the following text, remember no spaces (or line feeds) before or after when you paste it.

```
https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/underage.png?v=1658408579517
```

- The following view shows the entry using the expanded view of the dialog as an example.



Variable Name: imgUnderAge (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/underage.png?v=1658408579517

5 Click **+ Field** button to the right of Set Variables, this will create a new line with Variable Name configure it as follows:

5.1 From the **dropdown** field select **imgRobotDetected** from the list to configure for this flow:

5.2 In the **Variable Value** field copy and paste the following text, remember no spaces (or line feeds) before or after when you paste it.

```
https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/robot-detected.png?v=1658408579586
```

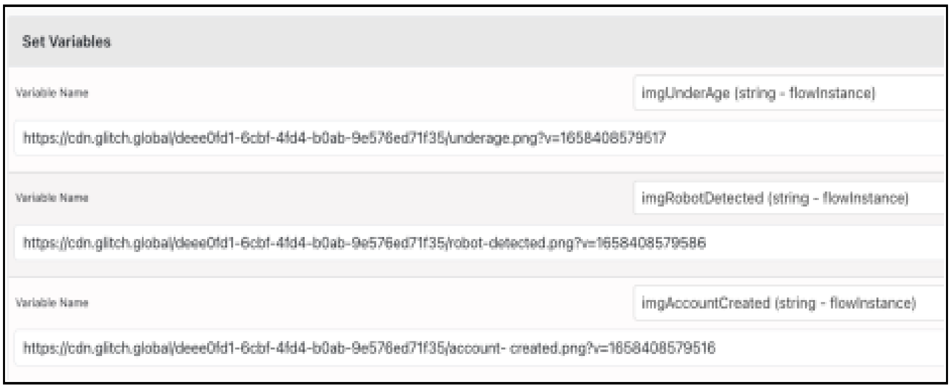
6 Click **+ Field** button to the right of Set Variables, this will create a new line with Variable Name configure it as follows:

6.1 From the **dropdown** field select **imgAccountCreated** from the list to configure for this flow:

6.2 In the **Variable Value** field copy and paste the following text, remember no spaces (or line feeds) before or after when you paste it.

```
https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/account-created.png?v=1658408579516
```

7 You will have three variables defined:



Set Variables

Variable Name: imgUnderAge (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/underage.png?v=1658408579617

Variable Name: imgRobotDetected (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/robot-detected.png?v=1658408579586

Variable Name: imgAccountCreated (string - flowInstance)

https://cdn.glitch.global/deee0fd1-6cbf-4fd4-b0ab-9e576ed71f35/account- created.png?v=1658408579516

8 Click **Apply** button to save your changes.

9 In the **Settings** tab enter the following:

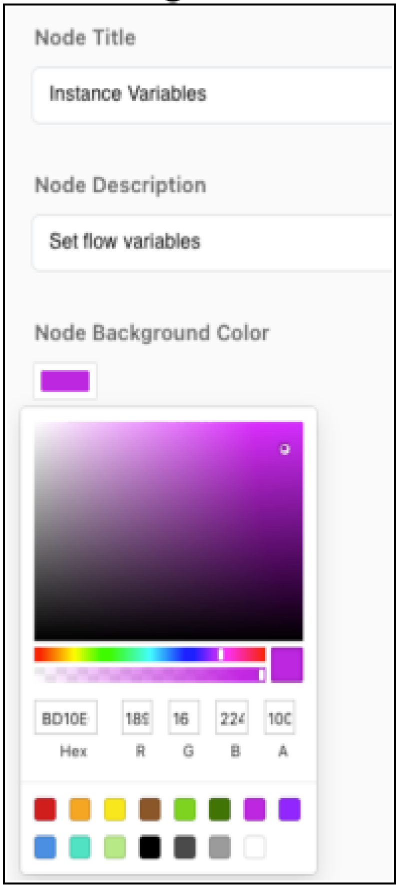
9.1 **Node Title** is

- **Instance Variables**

9.2 **Node Description** is

- **Set flow variables**

9.3 **Node Background Color** select color of lighter **purple** or **magenta**



Node Title: Instance Variables

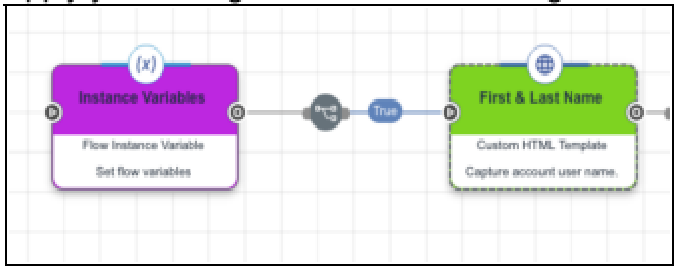
Node Description: Set flow variables

Node Background Color

Hex: BD10E
R: 189
G: 16
B: 224
A: 100

- You can select another color if you prefer the main idea is to use coloring coding to help identify the type of work your nodes are doing.
- This color coding should be something that is agreed on by everyone in your organization that is doing DaVinci flows.

10 Apply your changes and close the dialog.

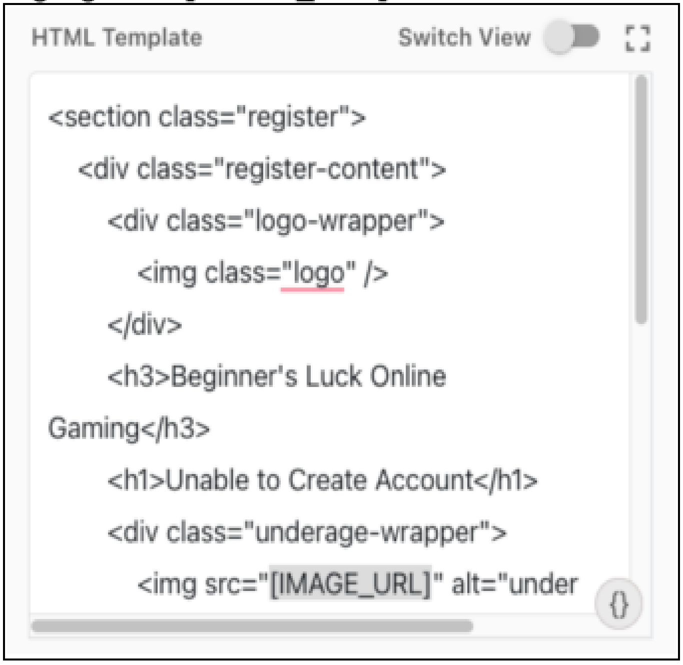


11 Continue with the next section.

2.1.2 Change HTTP message nodes to use variables

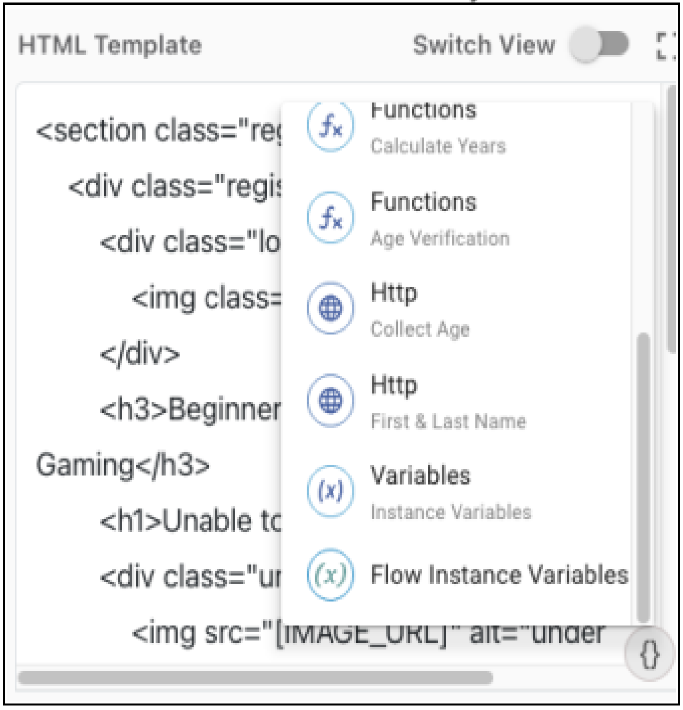
Now it is time to adjust the HTTP nodes that need to reference the image URLs to properly display the intended content to the user.

1 In your flow click **Not Legal Age** node.
2 In the **HTML Template** field scroll until you see the text **[IMAGE_URL]** this is the variable reference that needs to be replaced.
3 Highlight the **[IMAGE_URL]** text in the field.



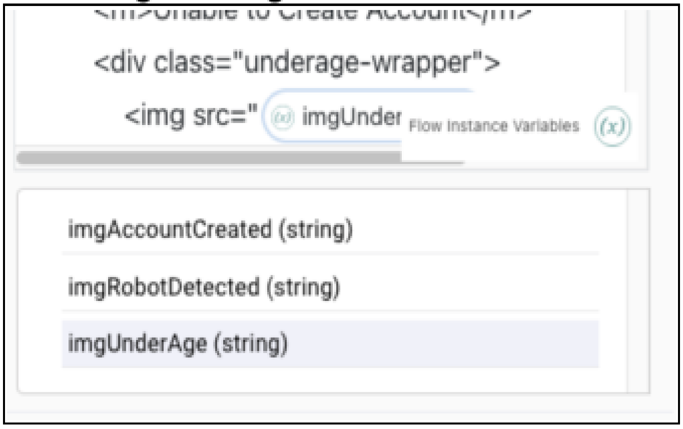
```
<section class="register">
    <div class="register-content">
        <div class="logo-wrapper">
            <img class="logo" />
        </div>
        <h3>Beginner's Luck Online
Gaming</h3>
        <h1>Unable to Create Account</h1>
        <div class="underage-wrapper">
            <img src="[IMAGE_URL]" alt="under
```

4 Click **{}** icon and scroll down to where you see **Flow Instance Variables**, you want to select this rather than the Variables node as you have done in the past.



```
<section class="re
    <div class="regis
        <div class="lo
            <img class=
        </div>
        <h3>Beginner
Gaming</h3>
        <h1>Unable to
        <div class="u
            <img src="[IMAGE_URL]" alt="under
```

5 Select **Flow Instance Variables** node.
6 Select **imgUnderAge** variable from the list.



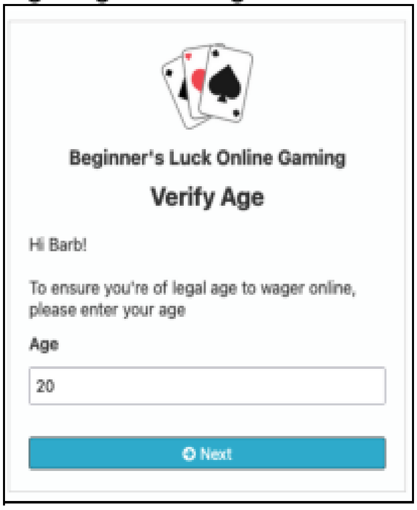
```
        <h1>Unable to Create Account</h1>
        <div class="underage-wrapper">
            <img src=" imgUnder
```

7 Apply your changes and close the dialog.
8 Good idea to save your flow at this point and anytime you make changes to the nodes, if needed.
9 Click **Account Created** node.
10 In the **HTML Template** field scroll until you see the text **[IMAGE_SUCCESS]** this is the variable reference that needs to be replaced.
11 Highlight the **[IMAGE_SUCCESS]** text in the field.
12 Select **Flow Instance Variables** node.
13 Select **imgAccountCreated** variable from the list.
14 Apply your changes and close the dialog.
15 Click **Robot Test Failed** node.
16 In the **HTML Template** field scroll until you see the text **[IMG_ROBOT_DECTED]** this is the variable reference that needs to be replaced.
17 Highlight the **[IMG_ROBOT_DECTED]** text in the field.
18 Select **Flow Instance Variables** node.
19 Select **imgRobotDetected** variable from the list.
20 Apply your changes and close the dialog.
21 Continue with the next section.

2.1.2.1 Test that images display

Now it is time to test the application to verify that the variables were used and the proper images were displayed on the page to the user.

1 Save and deploy your changes then **Try Flow** button.
2 Enter a user name and at the collect age page enter a value less than **21** to generate the not legal age message.



Beginner's Luck Online Gaming

Verify Age

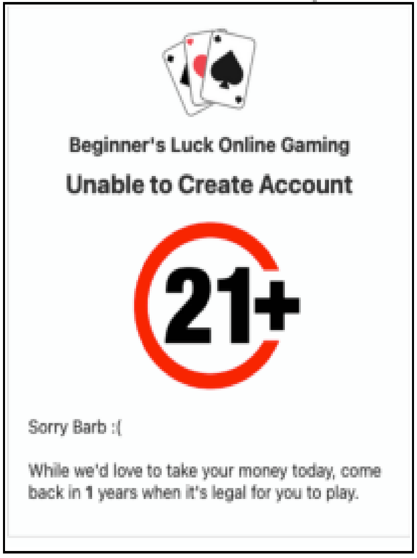
Hi Barb!

To ensure you're of legal age to wager online, please enter your age

Age

20

3 Click **Next** button and your dialog should have an image like the following screenshot:



Beginner's Luck Online Gaming

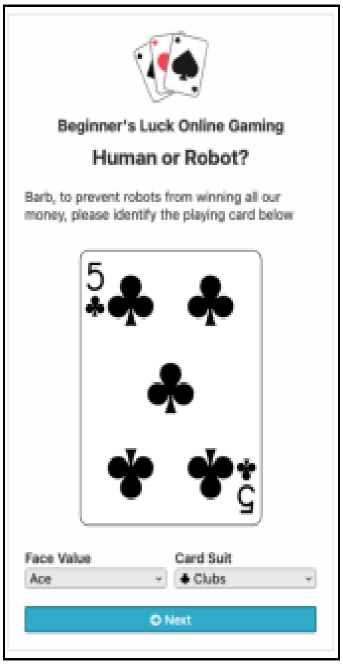
Unable to Create Account

Sorry Barb :(

While we'd love to take your money today, come back in **1** years when it's legal for you to play.

4 You cannot continue so close this browser tab and return to your flow.
5 Click **Try Flow** button.
6 Enter name, an age over 21, email, and password until you get to the robot test.

7 Select an invalid value or suit so that the test will fail.



Beginner's Luck Online Gaming

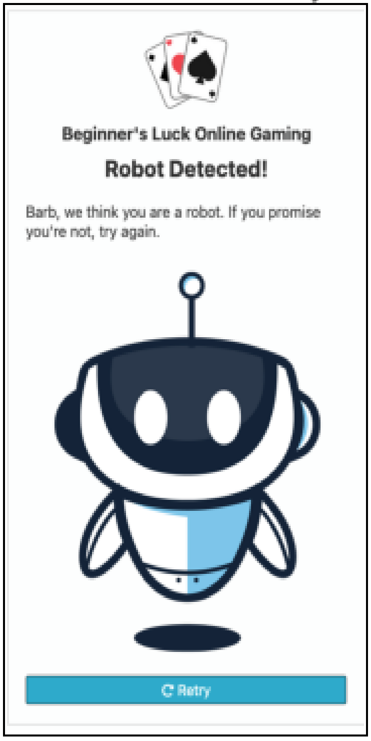
Human or Robot?

Barb, to prevent robots from winning all our money, please identify the playing card below

Face Value: Ace

Card Suit: ♣ Clubs

8 Click **Next** button and your dialog should have an image like the following screenshot:



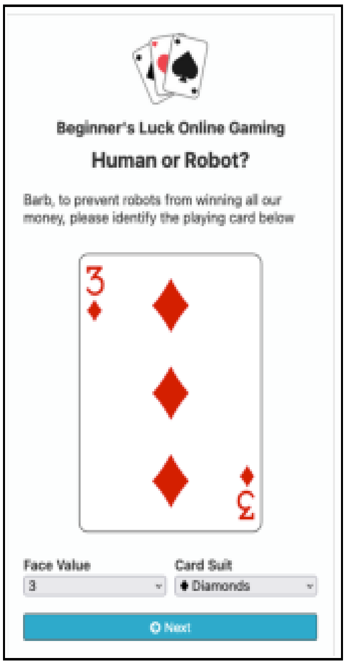
Beginner's Luck Online Gaming

Robot Detected!

Barb, we think you are a robot. If you promise you're not, try again.

9 Click **Retry** button to run the test again and see the final page image.

10 Select a valid value and suit so that the test will pass.



Beginner's Luck Online Gaming

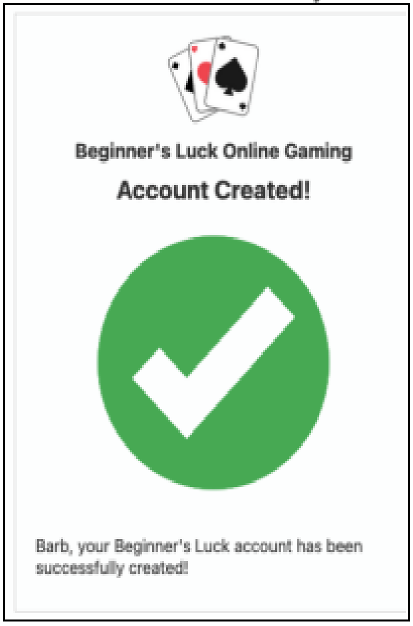
Human or Robot?

Barb, to prevent robots from winning all our money, please identify the playing card below

Face Value: 3

Card Suit: ♦ Diamonds

11 Click **Next** button and your dialog should have an image like the following screenshot:



Beginner's Luck Online Gaming

Account Created!

Barb, your Beginner's Luck account has been successfully created!

12 You have completed the test and may close the browser tab.

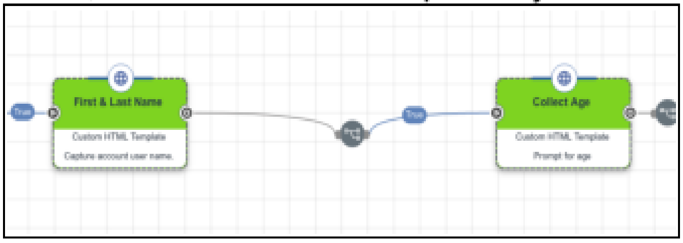
13 Continue with the next section.

2.1.3 Add variable for user first name

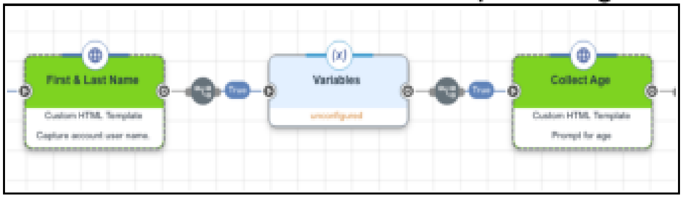
In this section you will define a variable node to hold information about the user that could be used in later nodes, rather than referencing the original node. This is done by placing the value into a variable node using a Variables node but then you can reference it anywhere because it becomes a global variable to the flow and is placed in a special bucket.

You will not be dealing with all the user data and all the potential uses of the data in this exercise. The goal is more to demonstrate the concept. You will use the first name data and only change one node. Normally this is something you would define upfront as a base requirement and manage it all along in your flow.

1 You are going to add a new variable node in between **First & Last Name** and **Collect Age** nodes, therefore make some space in your flow.



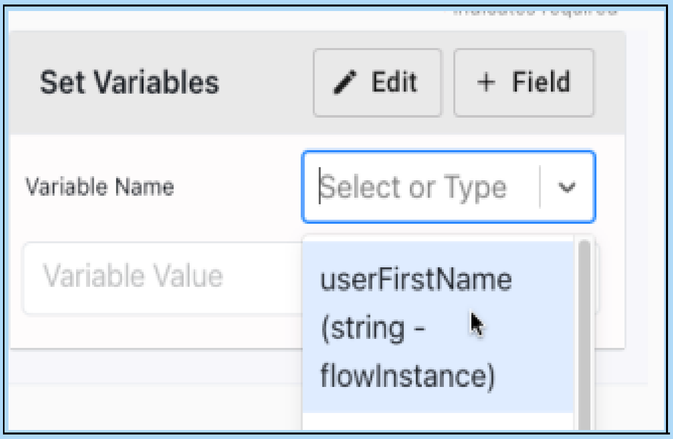
2 Delete the **Action Decision** node connecting these two nodes.
3 Add a new **Variables** connector in the space you made.
4 Connect the **Variables** node to its preceding and following nodes.



5 Click **Variables** node.
6 Click **Flow Instance Variable** capability.
7 Click **+ Field** button to the right of Set Variables

Note

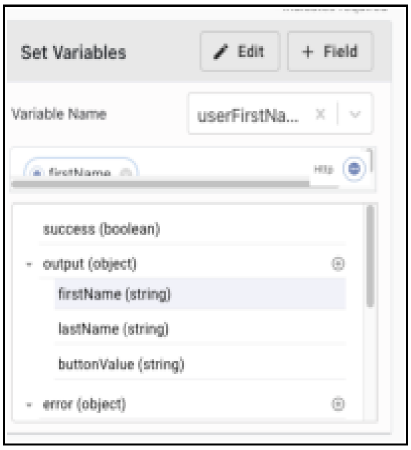
If you have already defined the variable in the Variables tab of the admin console in an earlier section in this flow then just select it from the list, rather than creating it.



8 **Variable Name** is:
- **userFirstName**

9 In the **Data Type** field select **String** from the dropdown list.
10 Accept the remaining default values.
11 Click **Create** button to create the variable and close the dialog.

12 **Value** is variable **firstName** from the **First & Last Name** node.



Set Variables

Variable Name: userFirstNa...

firstName

13 In the **Settings** tab enter the following:

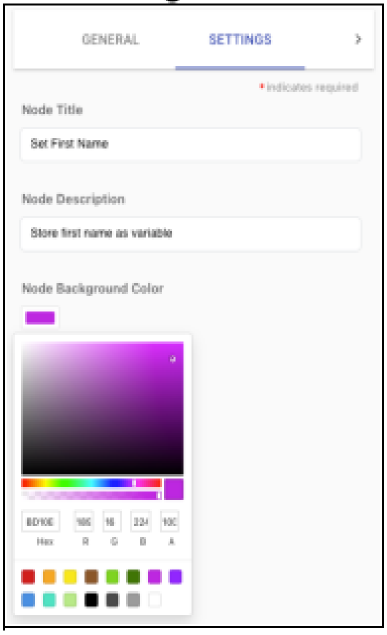
13.1 **Node Title** is

- **Set First Name**

13.2 **Node Description** is

- **Store first name as variable**

13.3 **Node Background Color** select color of lighter **purple** or **magenta**



* indicates required

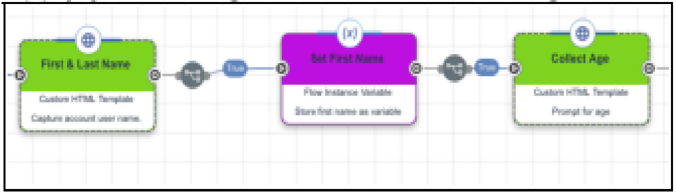
Node Title: Set First Name

Node Description: Store first name as variable

Node Background Color

Hex	R	G	B	A
BD10E	185	16	224	100

14 Apply your changes and close the dialog.

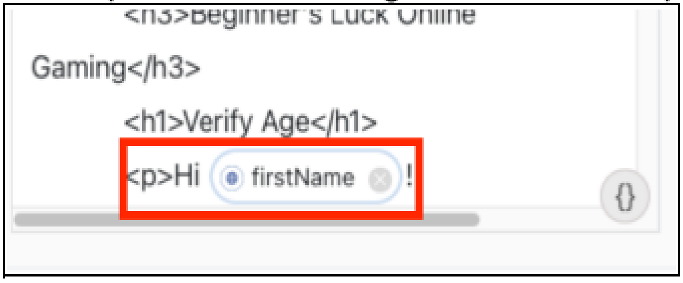


15 Continue with the next section.

2.1.4 Use the variable in Collect Age

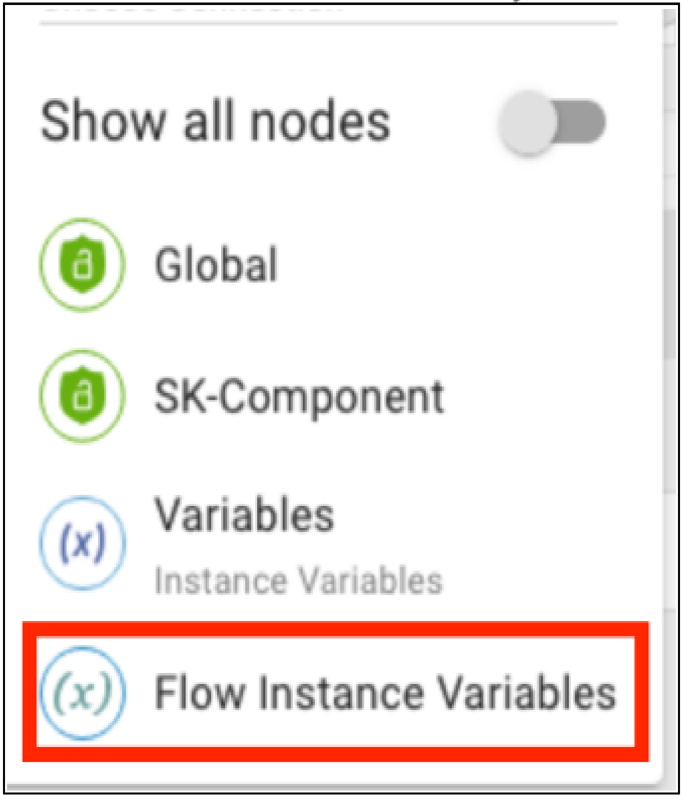
In this section you will change one node to use the variable instead of the direct reference back to the originating node. You can do this for all the nodes that reference `firstName` but only one example is needed.

1. Click **Collect Age** node.
2. In the **HTML Template** field scroll until you see the text **firstName** variable reference that needs to be replaced. The following screenshot will help you identify what you are looking for:



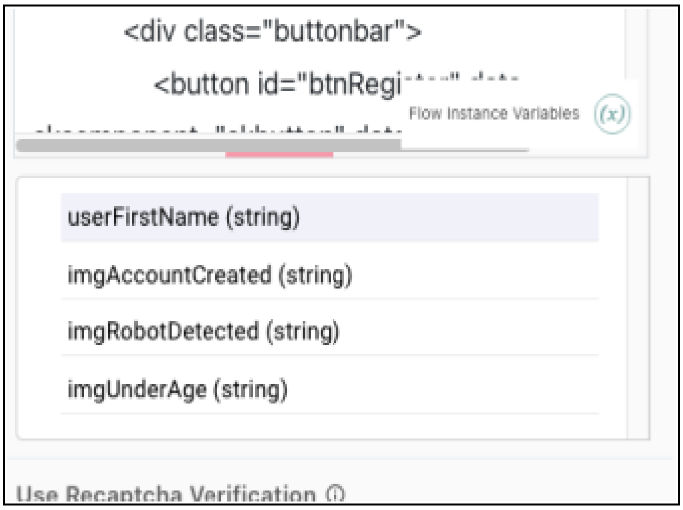
```
        <h3>Beginner's Luck Online
Gaming</h3>
        <h1>Verify Age</h1>
        <p>Hi firstName !
```

3. Click the **x** to the right of `firstName` to remove it.
4. Leave your mouse cursor before the **!** in the template this is where you want to place the new variable.
5. Select **Flow Instance Variables** entry, not the Variables node.



   - instance variable references are global objects to the flow and should be referenced from the Flow Instance Variables entry instead of the node that it is set in.

6. Select **userFirstName** variable from the list.



```
        <div class="buttonbar">
          <button id="btnRegi
```

userFirstName (string)

imgAccountCreated (string)

imgRobotDetected (string)

imgUnderAge (string)

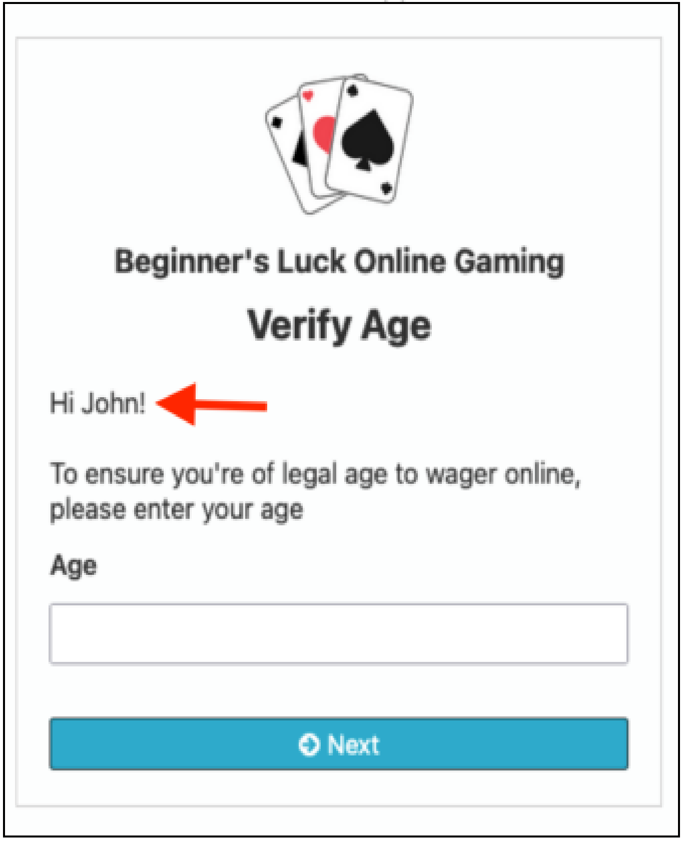
7. Apply your changes and close the dialog.

8 Continue with the next section.

2.1.4.1 Test your change that Collect Age page will show first name

Now it is time to test your change in using the variable, will the first name still appear on the Collect Age page.

1 Save and deploy your changes then **Try Flow** button.
2 Enter a first and last name and continue to the Collect Age page.
3 The first name should still appear after the text **Hi**



Beginner's Luck Online Gaming

Verify Age

Hi John!

To ensure you're of legal age to wager online, please enter your age

Age

4 Continue with the next section.

2.2 Form validation

So far, the only validation check you included in the registration flow was verifying that the user was over 21 years old. Typically you will have required fields or entry limitations, and the flow will need to prevent the user from proceeding if incorrect information is entered.

In implementing validation there ae two things to consider; how the entry will be checked and how the message will be handled. In this exercise you will take two approaches to this in following sections. Again the validation is going to be basic and not complete for all use cases, the goal is to show you the options.

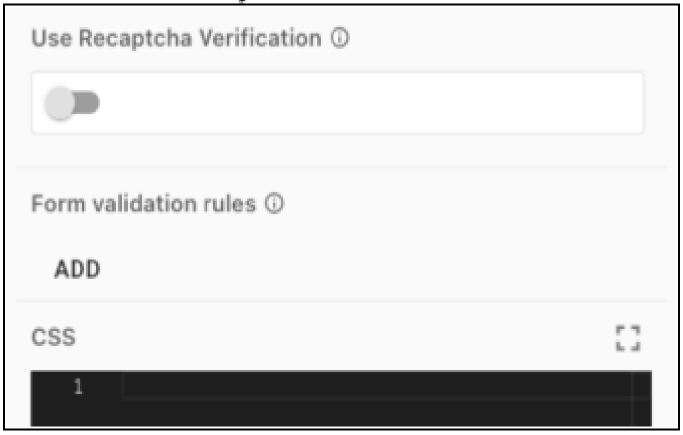
1 Make sure you have save your flow to this point.
2 Optionally you may want to clone the flow as a backup.
3 Continue with the next section.

2.2.1 Validate entry with property rules

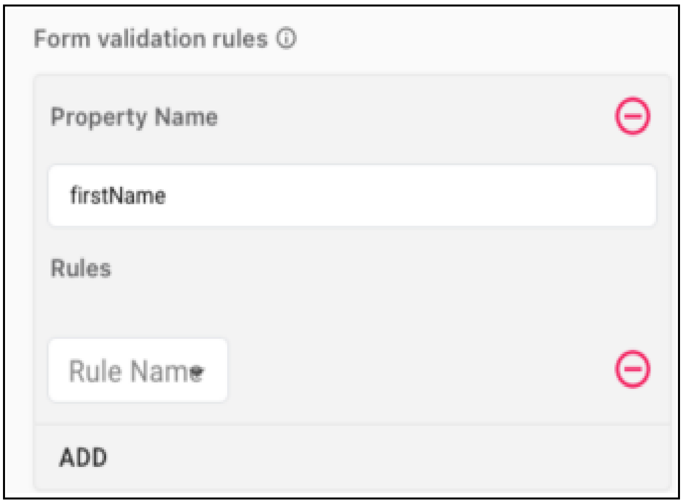
In this section you will require that the user enter a first name when that information is collected from the user. In this case you will use a rule to the property to validate the entry. The message will be displayed to the user on the same form using some built in DaVinci controls in HTML.

This approach is for simpler validation requirements on data input.

1 Click **First & Last Name** node.
2 Scroll down until you see **Form validation rules** section in the dialog.



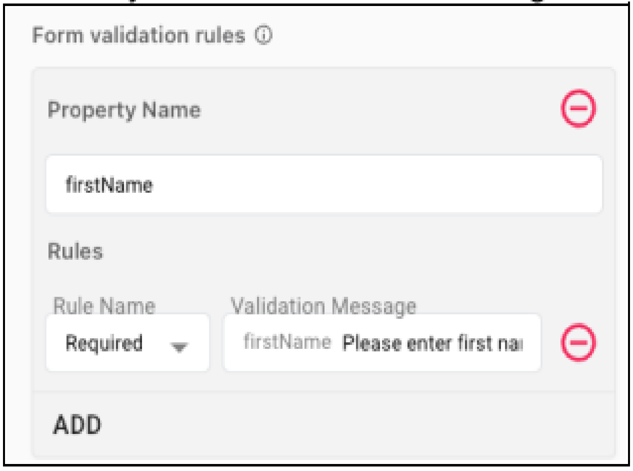
3 Click **ADD** button under **Form validation rules** to add a rule.
4 In **Property Name** enter **firstName**
5 Click **ADD** button under **Rules** to add a rule.



Property Name: firstName

6 In the **Rule Name** dropdown select **Required**
   - Note the other options for basic validation of input of a field.
7 In the **Validation Message** field after firstName enter the following text:
   - **Please enter first name**

8 Your entry should look like the following screenshot at this point:



Form validation rules ⓘ

Property Name

firstName

Rules

Rule Name	Validation Message
Required	Please enter first nai

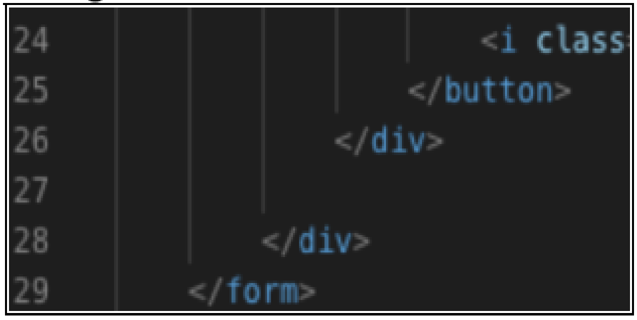
9 Click **Apply** button to save your changes to the node configuration to this point.

10 In order to display the message to the user you must change the HTML template using the following steps.

10.1 In the **HTML Template** toggle **Switch View** to change to the editor to make the code changes required.

10.2 Expand the view of this field to take the entire canvas view in your browser.

10.3 You will adding an HTML block to the existing template. This block needs to be after the **div tag** for the buttonbar div.



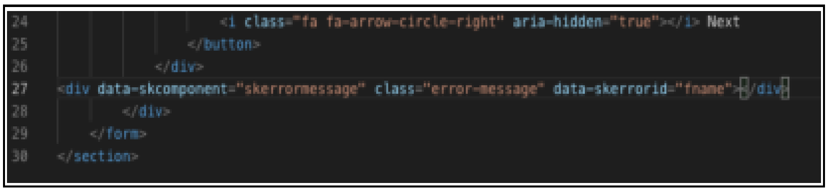
```
                <i class
            </button>
        </div>

    </div>
</form>
```

10.4 **Copy and paste** the following HTML line in the blank line after end of the buttonbar div element, as shown above.

```
<div data-skcomponent="skerrormessage" class="error-message"
    data-skerrorid="firstName"></div>
```

- You can use the format option of the right-mouse context menu in the editor to make it look nice.
- This is a DaVinci component of type skerrormessage. To use it, we must ensure the data-skerroridid matches the id from our form field.



```
                <i class="fa fa-arrow-circle-right" aria-hidden="true"></i> Next
            </button>
        </div>
<div data-skcomponent="skerrormessage" class="error-message" data-skerrorid="fname"></div>
    </div>
</form>
</section>
```

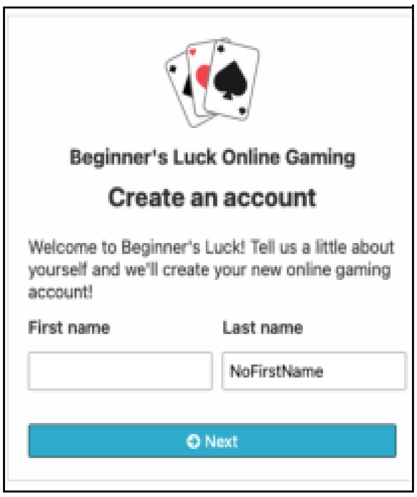
11 Apply your changes and close the dialog.

12 Continue with the next section.

2.2.1.1 Test validation changes for first name

Now it is time to test the form validation changes you just placed in your flow.

1. Save and deploy your changes then **Try Flow** button.
2. Don't enter a first name in the field.



Beginner's Luck Online Gaming

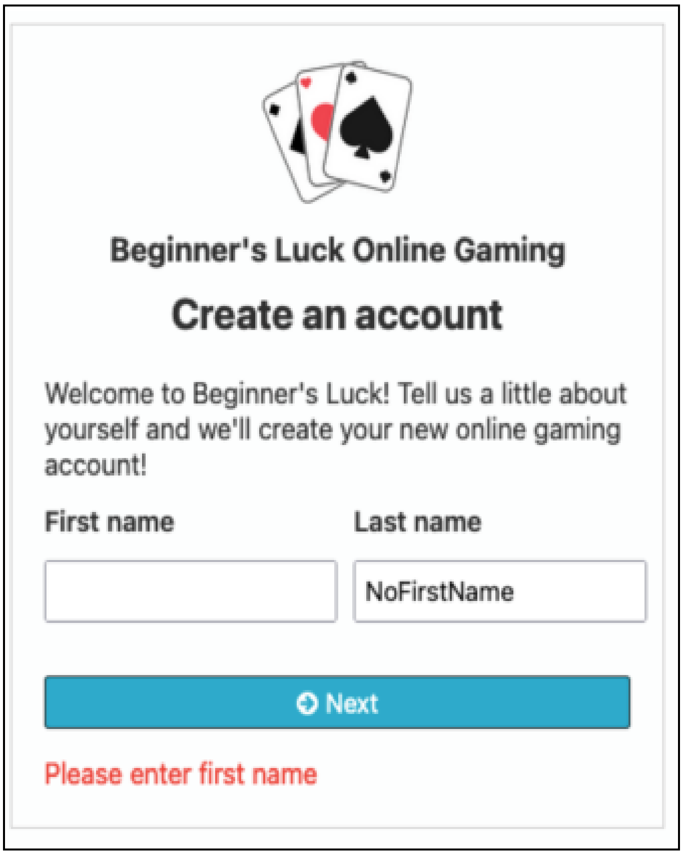
Create an account

Welcome to Beginner's Luck! Tell us a little about yourself and we'll create your new online gaming account!

First name:

Last name: NoFirstName

3. You can enter something in the last name field or not.
4. Click **Next** button



Beginner's Luck Online Gaming

Create an account

Welcome to Beginner's Luck! Tell us a little about yourself and we'll create your new online gaming account!

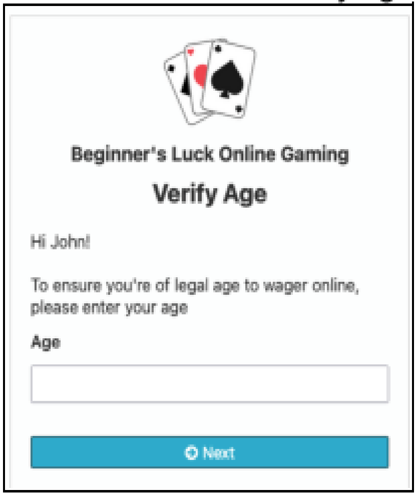
First name:

Last name: NoFirstName

Please enter first name

   - Just tabbing out of the field may render the message.

5. Enter a first name and try again.



Beginner's Luck Online Gaming

Verify Age

Hi John!

To ensure you're of legal age to wager online, please enter your age

Age

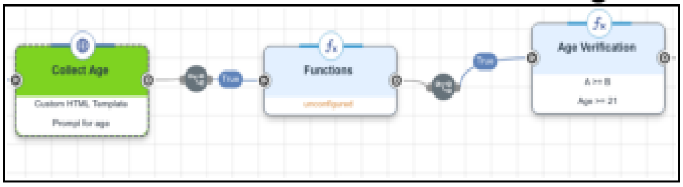
6. You can close the browser tab with your test page.
7. Continue with the next section.

2.2.2 Validate entry with functions and use error message component

In this section you will require that the user enter a valid age within a particular range when that information is collected from the user. In this case you will use a Functions connector to validate the entry. The function node will then route to an Error Connector to handle the message and display. The message will be displayed to the user on the same form using a different (then used in previous section) built in DaVinci control in HTML.

The Error Message connector and component approach is for more involved validation requirements for example checking across different entries or in this case a range.

1. You will need to make some room in your flow between the **Collect Age** and **Age Verification** nodes in your flow.
2. Highlight from the **Age Verification** node all the way to the end of the flow to drag it over some for another node.
    - You can use the keyboard cursor buttons to move things over or you can reduce the size of the flow to see all the nodes.
    - Use whatever method is easier for you.
3. Delete the **Action Decision** node between **Collect Age** and **Age Verification** nodes.
4. Click and drag a line from **Collect Age** node and release the mouse button.
5. Click on **Functions** connector to add a new node.
6. Connect the new **Functions** node to the **Age Verification** node.



7. Click the new **Functions** node.
8. Select **B < A < C** capability from the list.
    - We want age to be in the range of 1 to 120, this will make a positive integer within a valid

age range.

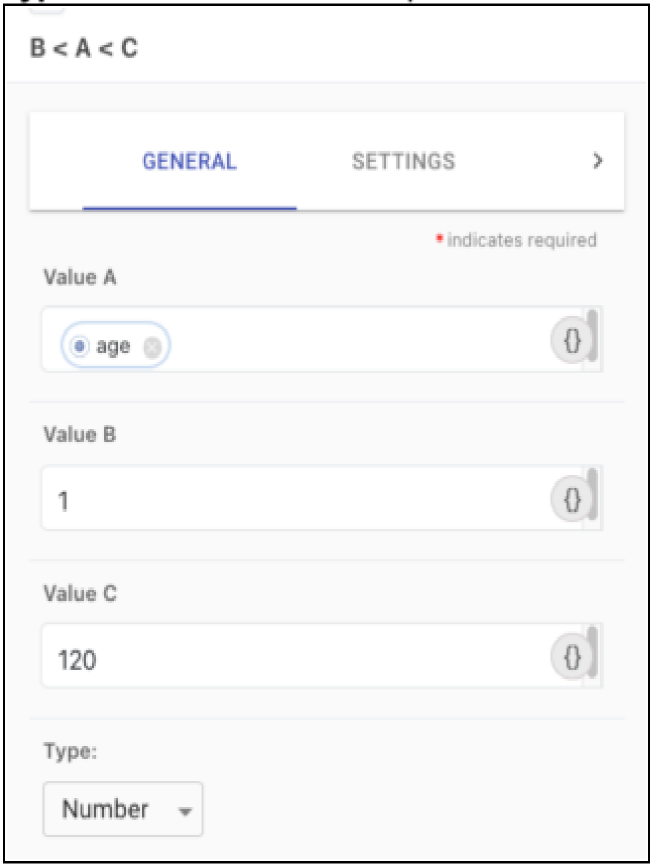
9 Configure the capability as follows in the **General** tab.

9.1 **A** is the variable **age** from the **Collect Age** node.

9.2 **B** is **1**

9.3 **C** is **120**

9.4 **Type** is **Number** from the dropdown list



B < A < C

Value A: age

Value B: 1

Value C: 120

Type: Number

10 Select **Settings** tab and enter

10.1 **Node Title** is

- **Valid Age Check**

10.2 **Node Description** is

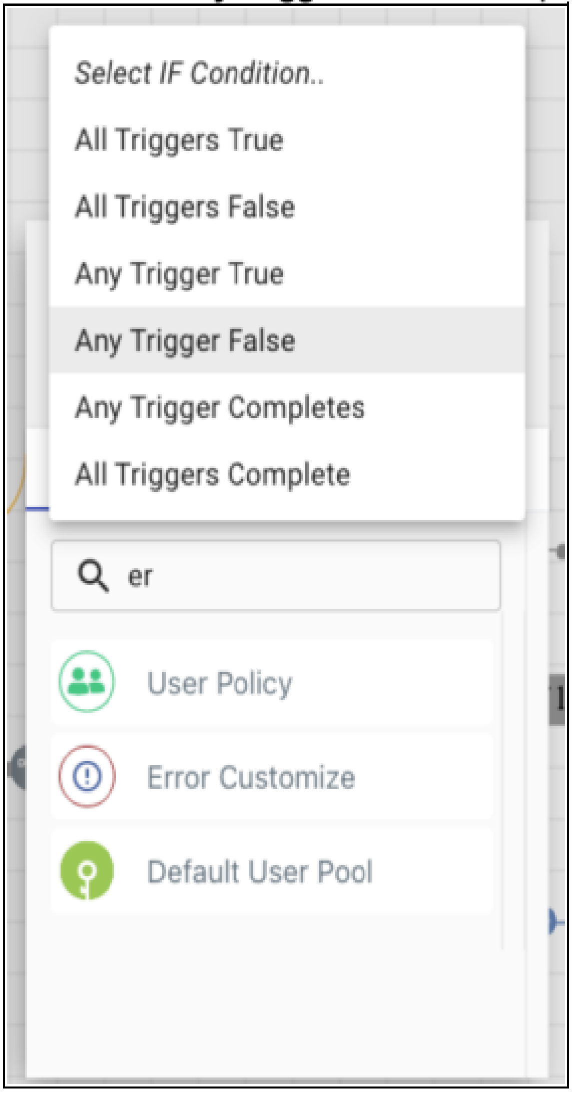
- **Age between 1 and 120**

11 Apply your changes and close the dialog.

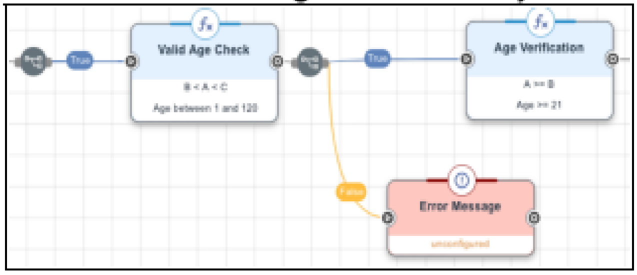
12 And don't forget to save your changes to the flow.

13 From the **Action Decision** node after **Valid Age Check** node drag a line and release the mouse button.

13.1 It should be **Any Trigger False** for the path.



13.2 Add an **Error Message** connector to your flow.

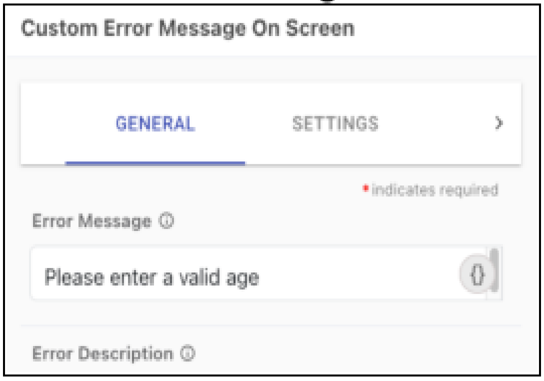


13.3 Click the **Error Message** node and configure as follows:

13.3.1 Click **Custom Error Message on Screen** capability

13.3.2 **Error Message** as

- **Please enter a valid age**



Custom Error Message On Screen

GENERAL SETTINGS

* indicates required

Error Message: Please enter a valid age

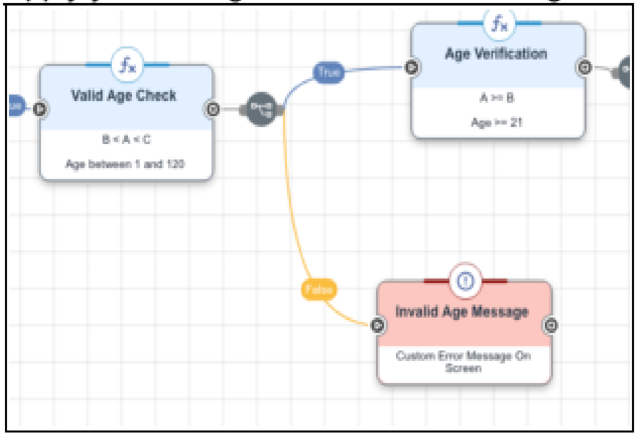
Error Description

13.3.3 Click **Settings** tab

13.3.4 **Node Title** is

- **Invalid Age Message**

13.4 Apply your changes and close the dialog.

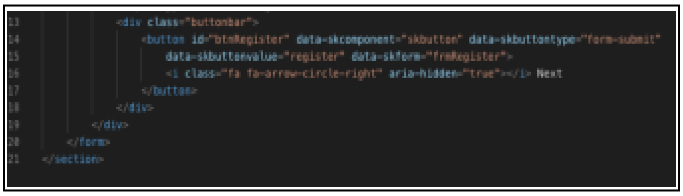


14 Continue with the next section.

2.2.2.1 The template for collect age needs component to display the message

Now you need to add the DaVinci component to the age page so that the message is displayed if the user does not enter a valid age. The message will be displayed on the last page presented to the user, which in this case is the Collect Age node.

1 Select **Collect Age** node to edit it.
2 In the **HTML Template** toggle **Switch View** to change to the editor to make the code changes required.
3 Expand the view of this field to take the entire canvas view in your browser.
4 You will adding an HTML block to the existing template. This block needs to be after the **div tag** for the buttonbar div.

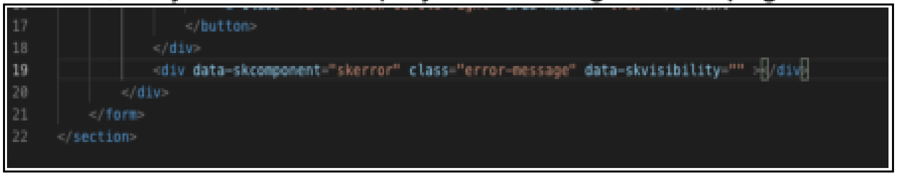


```
            <div class="buttonbar">
                <button id="btnRegister" data-skcomponent="skbutton" data-skbuttontype="form-submit"
                    data-skbuttonvalue="register" data-skform="frmRegister">
                    <i class="fa fa-arrow-circle-right" aria-hidden="true"></i> Next
                </button>
            </div>
        </div>
    </form>
</section>
```

5 Insert a blank line to copy the following text into it.
6 **Copy and paste** the following HTML line in the blank line after end of the buttonbar div element.

```
<div data-skcomponent="skerror" class="error-message" data-skvisibility="" ></div>
```

- This is a DaVinci component of type skerrormessage. In this case data-skerroridid is not needed, we just want to display the message on the page.



```
                </button>
            </div>
            <div data-skcomponent="skerror" class="error-message" data-skvisibility="" ></div>
        </div>
    </form>
</section>
```

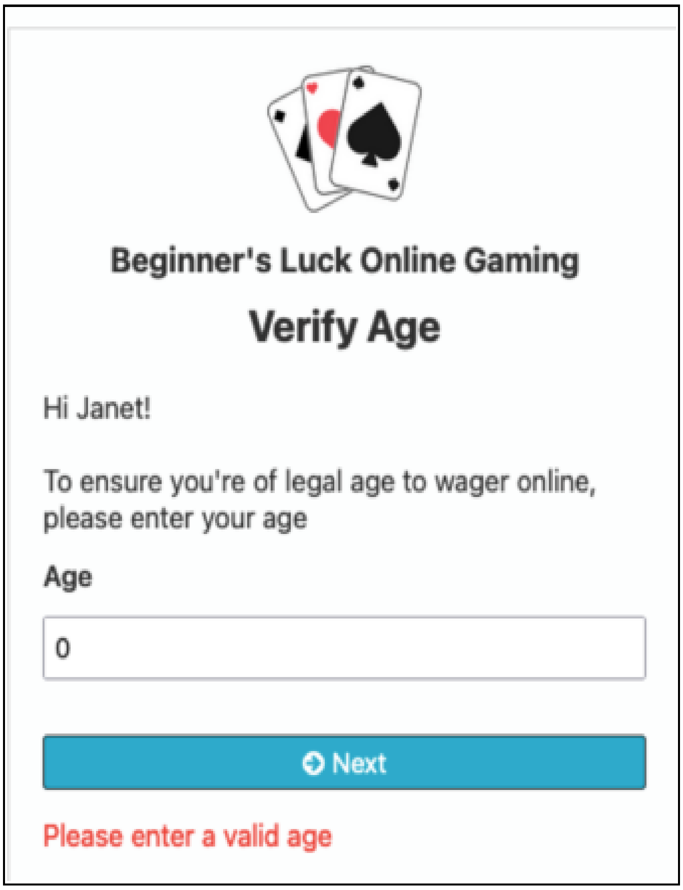
7 Apply your changes and close the dialog.
8 Continue with the next section.

2.2.2.2 Test the age validation check

Now it is time to check if the age validation works and presents the proper message at the correct form.

During testing if you get a timeout error, especially in the second use case just try again. Just a brief hiccup as you rapidly test your use cases.

1 Deploy your changes then **Try Flow** button.
2 Enter a first and last name and continue to the Collect Age page.
3 Enter **0** for the age.
4 Click **Next** button.



Beginner's Luck Online Gaming

Verify Age

Hi Janet!

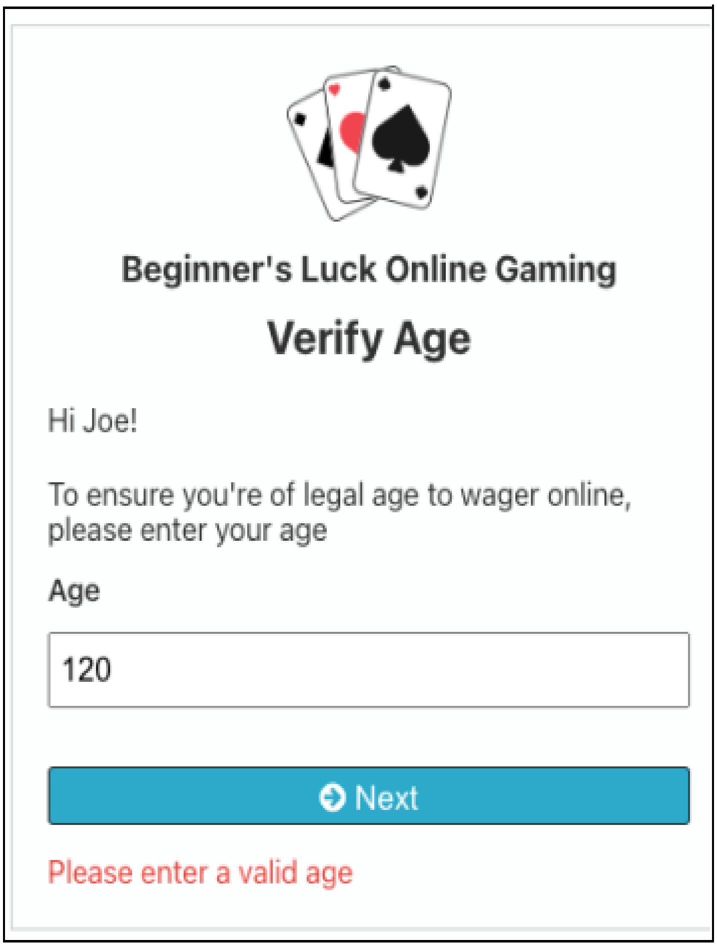
To ensure you're of legal age to wager online, please enter your age

Age: 0

Please enter a valid age

5 Enter **120** for the age.

6 Click **Next** button.



Beginner's Luck Online Gaming

Verify Age

Hi Joe!

To ensure you're of legal age to wager online, please enter your age

Age

120

Please enter a valid age

7 Enter **22** for the age.

- You will be able to proceed to the next page.

8 You can close the test browser tab as your test is complete.

9 You have completed this exercise.