

Convert User Interactions to Use HTML Templates

Table of Contents

- 1 Objective
- 2 Do the following task items
 - 2.1 Change user interaction nodes to use HTML templates
 - 2.1.1 Edit node First & Last Name
 - 2.1.1.1 Test your UI to this point
 - 2.1.1.2 Apply a CSS style sheet to flow
 - 2.1.1.3 Review the UI at this stage
 - 2.1.1.4 Using JavaScript to set focus
 - 2.1.2 Edit node Collect Age
 - 2.1.2.1 Test changes to Collect Age node
 - 2.1.2.2 Fix the variable reference in template
 - 2.1.2.3 Change age verification to use new variable
 - 2.1.3 Edit node Collect Email
 - 2.1.3.1 Test changes to Collect Email node
 - 2.1.4 Edit node Review Account Info
 - 2.1.4.1 Test changes to Review Account Info node
 - 2.1.5 Edit node Not Legal Age
 - 2.1.5.1 Test changes to Not Legal Age node
 - 2.1.6 Edit node Robot Test
 - 2.1.6.1 Test changes to Robot Test node
 - 2.1.7 Edit node Account Created
 - 2.1.8 Edit node Robot Test Failed
 - 2.2 Test the complete flow changes

1 Objective

In this exercise you will start using more advanced concepts in PingOne DaVinci to implement your flows. You will start with the flow from the previous exercise and begin to customize it. Your objective is to move from the basic UI used with the basic components to a richer UI that provides you more control over the style and user interaction options to make it a richer experience for the user of the application.

In the following exercises you will continue down this path to leverage even more advanced

concepts and best practices to improve your flows. Such as leveraging subflows and variables.

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

In this exercise you will be replacing all the user interaction and presentation nodes to use HTML templates that will leverage CSS and JavaScript for an enriched experience. In order to complete this exercise you will add HTML, JavaScript, and use CSS as you replace the existing nodes. The code that you need will be provided in the exercise for you to copy and paste in the correct blocks. As you do this copy and paste, review closely what you have in your flow with what is in the exercise to make sure everything will work as expected.

You are of course welcome to review the code and even ask the instructor about it, but ultimately the goal of this training is not about HTML, CSS, or JavaScript but about building user experiences with DaVinci for your applications.

2.1 Change user interaction nodes to use HTML templates

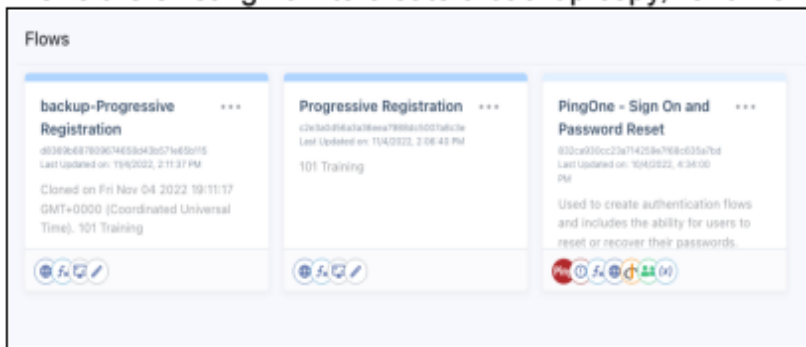
Prior to starting the conversion, let's run through the flow to see where we left it off. Presently, we have a progressive registration flow that captures user profile data, ensures they're of legal age, then renders a custom captcha test to verify they're a real user.

The basic outline of the steps you will follow to convert the UI are:

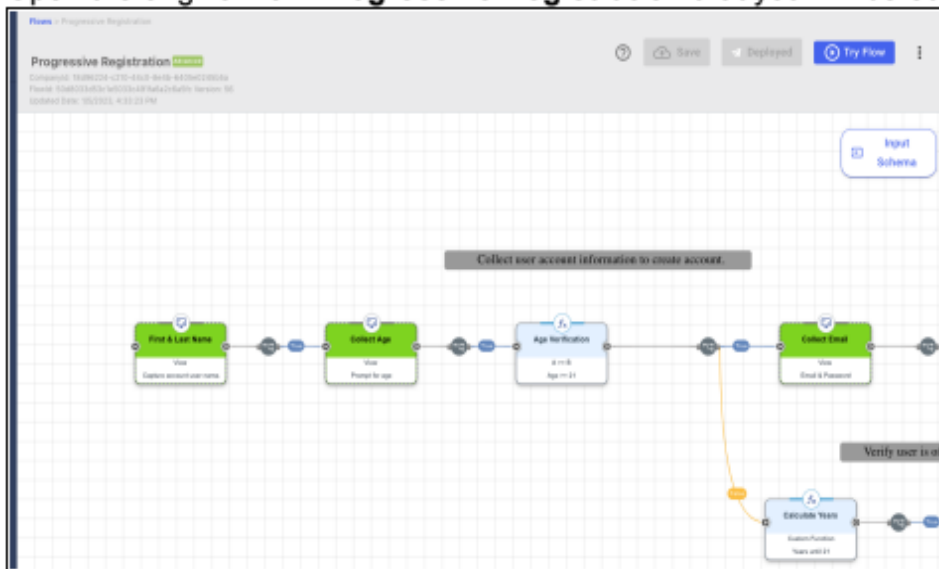
- To migrate an existing node that provide interaction with with user to custom HTML, will depend on the type of node:
 - For Screen type nodes you will replace the node with a new Http node, redefining the properties that are inputs on the existing form.
 - For Http type nodes you edit the component, then on the Trigger, and select Custom HTML Template and add the new presentation layer, mapping variables to the new template.
- For each edit of the new or existing component you will copy new HTML, ensuring the form input fields have identical IDs to the component's properties.
- DaVinci has a number of components and types that you can use within your HTML. Here, we're using a form submit button with a `data-skcomponent` type of `skbutton`. This will allow DaVinci to capture the values entered in the form.

1 You will be updating the flow **Progressive Registration** you completed in the previous exercise.

- 2 Clone the existing flow to create a backup copy, **rename** it as a backup.



- 3 Open the original flow **Progressive Registration** that you will be editing in this exercise.



- 4 Continue with the next section.

2.1.1 Edit node First & Last Name

First up you will change the first node with user interaction, where you collect the user name. Then after that you will tackle the remaining ones in order to adjust the presentation and interaction, testing each change as you go along.

In this section for the first Screen connector type node the instructions here will be pretty detailed. In the following sections where it is pretty much the same pattern, instructions will not be as detailed.

You will start by adding the new node not connected to anything and configuring it out. This will allow you to review the existing node for configuration details if needed. Once done you will then add the new node to the flow and remove the old node.

- 1 Click **+** at the bottom left of the canvas to add a new **Http** node.
- 2 You may need to move it down a bit on the canvas if it gets dropped over the existing flow.
- 3 Click on the **Http** node that you just placed on the canvas to edit it.
- 4 Select the **Custom HTML Template** capability from the list.

4.1 Next to **HTML Template** field toggle **Switch View** on to change to the code editor for the field.



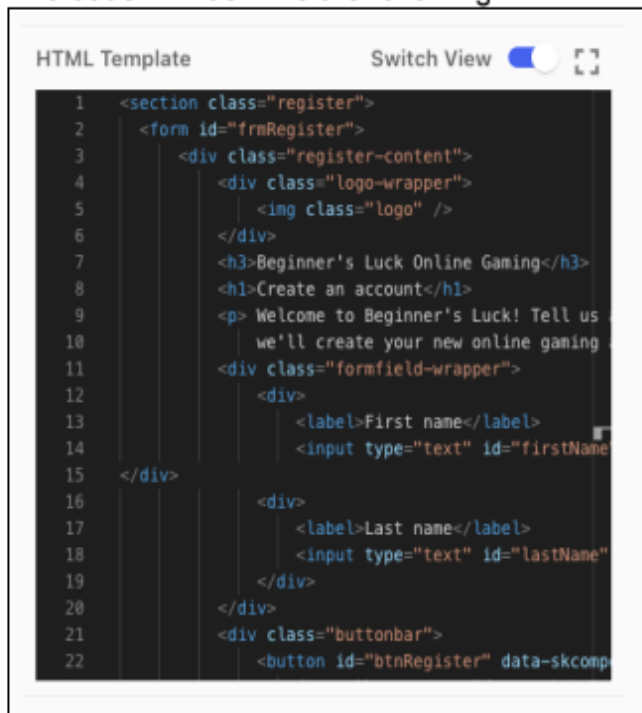
4.2 Copy the following block of code and paste it into the **HTML Template** field.

```
<section class="register">
  <form id="frmRegister">
    <div class="register-content">
      <div class="logo-wrapper">
        <img class="logo" />
      </div>
      <h3>Beginner's Luck Online Gaming</h3>
      <h1>Create an account</h1>
      <p> Welcome to Beginner's Luck! Tell us a little about yourself and
        we'll create your new online gaming account!</p>
      <div class="formfield-wrapper">
        <div>
          <label>First name</label>
          <input type="text" id="firstName" />
        </div>

        <div>
          <label>Last name</label>
          <input type="text" id="lastName" />
        </div>
      </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>
```


4.3 The code will look like the following:



```

1 <section class="register">
2   <form id="frmRegister">
3     <div class="register-content">
4       <div class="logo-wrapper">
5         <img class="logo" />
6       </div>
7       <h3>Beginner's Luck Online Gaming</h3>
8       <h1>Create an account</h1>
9       <p> Welcome to Beginner's Luck! Tell us
10         we'll create your new online gaming
11       <div class="formfield-wrapper">
12         <div>
13           <label>First name</label>
14           <input type="text" id="firstName"
15         </div>
16         <div>
17           <label>Last name</label>
18           <input type="text" id="lastName"
19         </div>
20       </div>
21       <div class="buttonbar">
22         <button id="btnRegister" data-skcomp

```

4.3.1 You can expand the field to full screen to see it all.

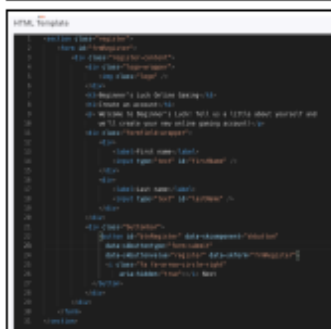
4.3.2 Some **div** tags may not align properly but in general that is okay.

4.3.3 If you **right-mouse** click in the editor you see **Format Document** menu option that can align your code all nice and neat.

4.3.3.1



4.3.3.2



4.4 Scroll down to the bottom to the **Output Fields List** to add a fields needed for the form.

- There are two defined in the form `firstName` and `lastName`
- If you look at the code above you will see that these are HTML `input` elements that have an `id` attribute with these names.

4.5 Click **Add** button to a new field.

4.5.1 Set the **Property Name** to `firstName`

- Case is important and must match what is in your template HTML.

4.5.2 The **Data Type** is **String**

4.5.3 You can take the defaults for everything else.

- The Value of course will be entered by the user

- Display name is already on the template HTML code as a `label`

4.6 Click **Add** button to a new field.

4.6.1 Set the **Property Name** to **lastName**

- Case is important and must match what is in your template HTML.

4.6.2 The **Data Type** is **String**

4.6.3 You can take the defaults for everything else.

- The Value of course will be entered by the user
- Display name is already on the template HTML code as a `label`

4.7 Click **Settings** tab at the top to name your node.

4.7.1 Set the **Node Title** to **First & Last Name**

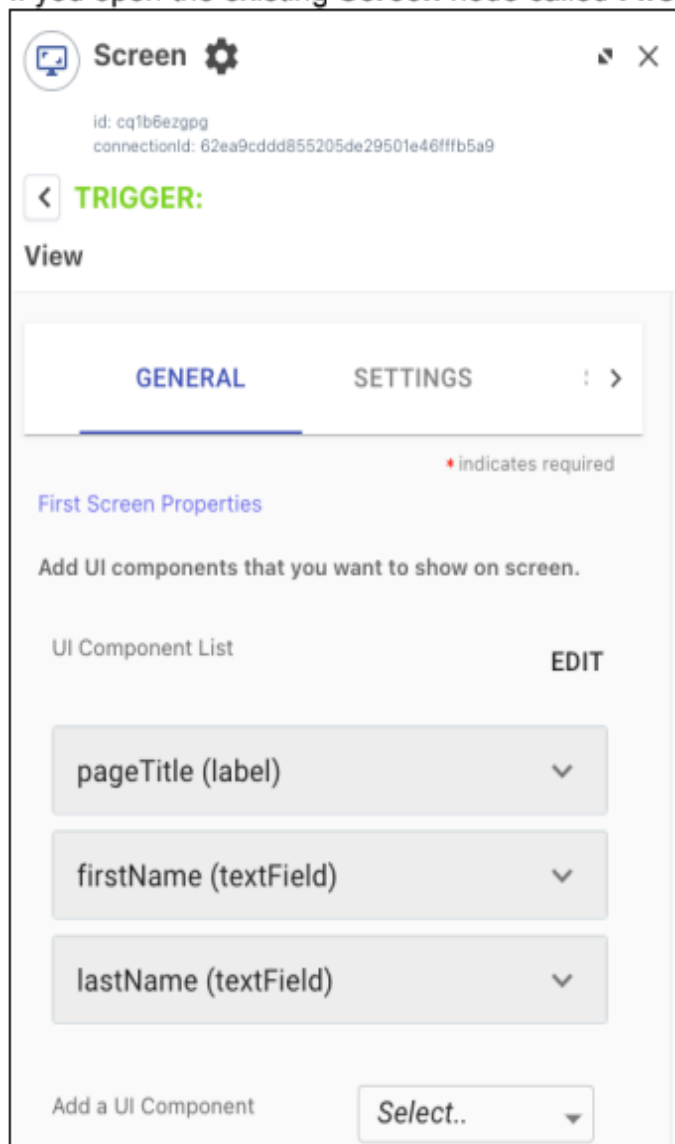
4.7.2 Set the **Node Description** to **Capture account user name.**

4.7.3 Set the **Node Background Color** to **lite green**

4.8 Apply and close your dialog.



5 If you open the existing **Screen** node called **First & Last Name** node.



- You have the two UI components that match your properties that you just added to the Http

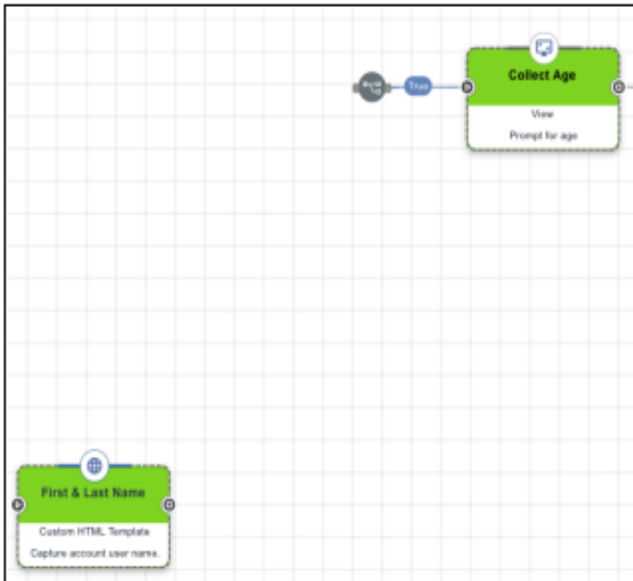
node.

- The label one is part of the template so you did not need to specify it.

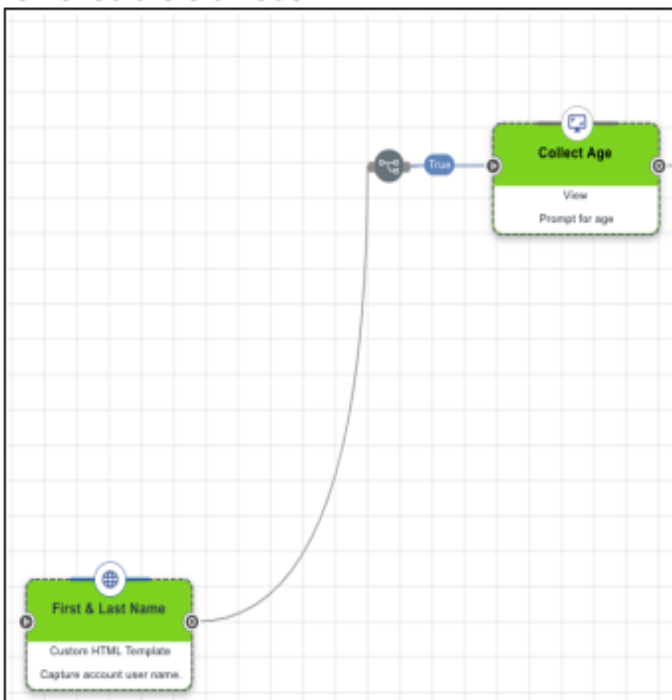
6 Close the dialog, making no changes.

7 **Right-mouse** click on the node you just reviewed, the **Screen** type node.

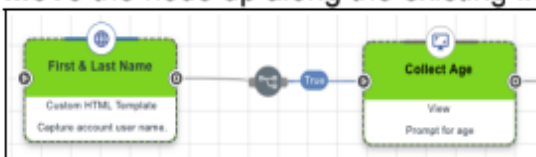
8 Select **Delete** menu item to delete the node.



9 From the new **First & Last Name** node connect it to the **Decision Action** node where you removed the old node.



10 **Move** the node up along the existing flow line to make your flow look neater.



11 Click **Save** button to save your flow.

- Another gentle reminder to this often as you edit a flow, it can be very frustrating if something happens in your browser your changes will be lost. When you try to close the flow you will be warned that you have unsaved changes.

12 Continue with the next section.

2.1.1.1 Test your UI to this point

Now it is time to try out the new template and see what the new UI looks like.


- 1 Click **Deploy** button to deploy your latest changes in preparation for testing.
- 2 Click **Try Now** button to test the latest changes.
 - You will only test to the first node not the entire flow.
 - You will follow this pattern a lot in this lab of only going as far as needed in your flow to see the changes you made and verify they are working.
- 3 In the new browser tab that was opened you will see the new template rendered.

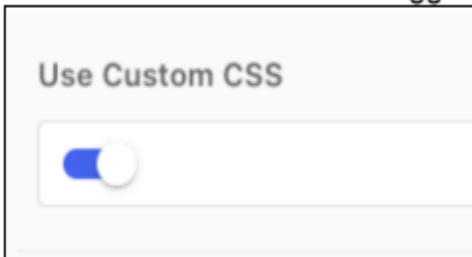


- 4 So the UI has changed, but it does look kind of basic.
- 5 **Close** the browser tab, no point in leaving all the test pages up when you are done.
- 6 Continue with the next section.

2.1.1.2 Apply a CSS style sheet to flow

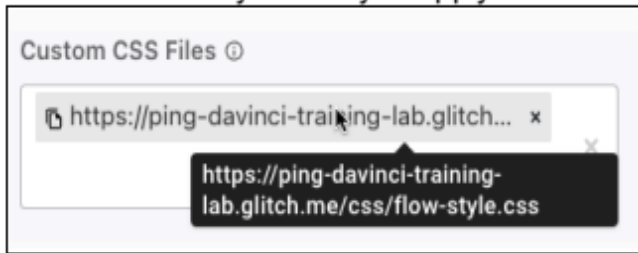
Your registration form doesn't look as good as before, so let's apply a CSS file to the flow. In future exercises, you'll move this to the flow's hosting application but for now, let's head into the settings menu and then customizations to add your CSS file.

- 1 Return back to the browser tab with your flow.
- 2 Click  icon after the **Try** button.
- 3 Click **Flow Settings** from the menu.
- 4 In the **Customizations** tab toggle the **Use Custom CSS** toggle on.



- 5 In the **Custom CSS Files** copy and paste **<https://ping-davinci-training-lab.glitch.me/css/flow-style.css>**
 - The above line must be pasted as one complete line with no spaces if it happens to wrap as you display the file.
 - If you are interested you can review the CSS that will be used: <https://ping-davinci-training-lab.glitch.me/css/flow-style.css>

6 Press **Return** key for entry to apply.



- You can enter multiple links if you had more than one CSS to use as separate entries.

7 Click **Save** button in the dialog, to save your changes to flow settings.

8 Click **Cancel** button close the dialog.

9 Continue with the next section.

2.1.1.3 Review the UI at this stage

Now it is time to review the template with the applied style sheet.

1 Save and deploy your flow then try it again.

2 Looking good!

3 Continue with the next section.

2.1.1.4 Using JavaScript to set focus

When you run the flow, notice that the focus isn't set on the first field in the form. If you were creating a standard HTML application, you would use the autofocus attribute on the form field.

Presently DaVinci doesn't honor this property. However, you can achieve the same result using JavaScript.

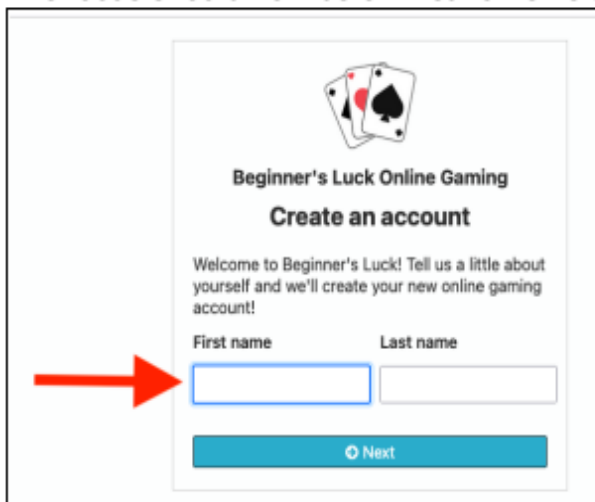
- 1 Click **First & Last Name** node in your flow.
- 2 **Copy** the following block of code.

```
document.getElementById("firstName").focus()
```

- 3 Then **paste** the code in the **Script** field.



- 4 Apply and close the dialog.
- 5 **Deploy** and try your changes.
- 6 The focus should now be on first name field on the form.



- 7 The changes to the first node and user interaction are now complete. You have seven more to complete in the following sections of this exercise.
- 8 Continue with the next section.

2.1.2 Edit node Collect Age

In editing this node the steps are pretty much the same as you did to edit the previous node. The instructions in this section will not be as detailed, it will just provide you general guidance as to what needs to be done.

- 1 Click **+** at the bottom left of the canvas to add a new **Http** node.
- 2 You may need to move it down a bit on the canvas if it gets dropped over the existing flow.
- 3 Click on the **Http** node the select the **Custom HTML Template** capability from the list.
 - 3.1 Next to **HTML Template** field toggle **Switch View** on to change to the code editor for the field.

3.2 Copy the following block of code and paste it into the **HTML Template** field.

```
<section class="register">
  <form id="frmRegister">
    <div class="register-content">
      <div class="logo-wrapper">
        <img class="logo" />
      </div>
      <h3>Beginner's Luck Online Gaming</h3>
      <h1>Verify Age</h1>
      <p>Hi [FNAME]!
      <p>To ensure you're of legal age to wager online,
        please enter your age</p>
      <label>Age</label>
      <input type="text" id="age" />
      <div class="buttonbar">
        <button id="btnRegister" data-skcomponent="skbutton"
          data-skbuttonontype="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>
```

3.3 Scroll down to the bottom to the **Output Fields List** to add a fields needed for the form.

- There is one defined in the form `age`
- If you look at the code above you will see it in the HTML `input` element with `id` attribute with the name.

3.4 Click **Add** button to a new field.

3.4.1 Set the **Property Name** to `age`

- Case is important and must match what is in your template HTML.

3.4.2 The **Data Type** is **Number**

3.4.3 You can take the defaults for everything else.

3.5 Copy the following block of code.

```
document.getElementById("age").focus()
```

3.6 Then **paste** the code in the **Script** field.

3.7 Click **Settings** tab at the top to name your node.

3.7.1 Set the **Node Title** to **Collect Age**

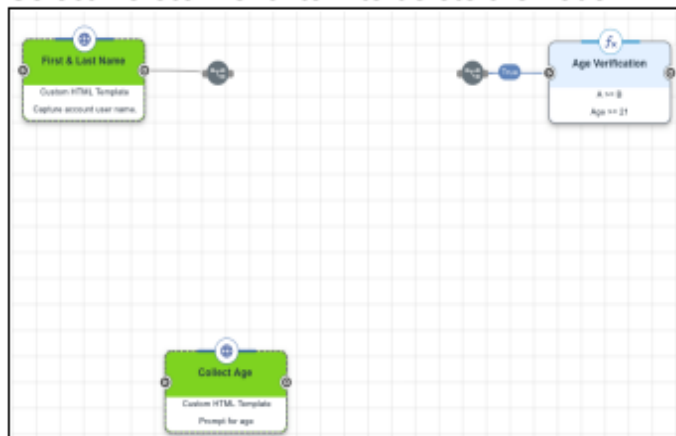
3.7.2 Set the **Node Description** to **Prompt for age**

3.7.3 Set the **Node Background Color** to **lite green**

3.8 Apply and close your dialog.



- 4 **Right-mouse** click on the node old screen node currently in the flow as **Collect Age** type node.
- 5 Select **Delete** menu item to delete the node.



- 6 Connect the new **Collect Age** node to hold the same spot in the flow as the old one.
- 7 **Move** the node up along the existing flow line to make your flow look neater.

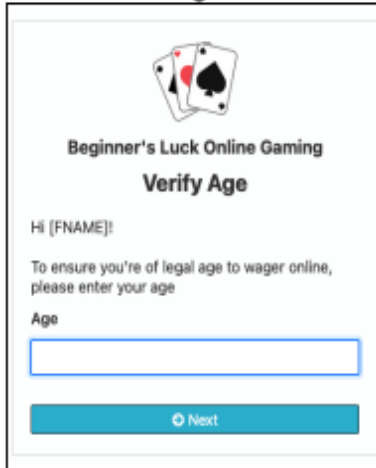


- 8 Click **Save** button to save your flow.
- 9 Continue with the next section.

2.1.2.1 Test changes to Collect Age node

Your changes are complete time to verify that they work. You will need to of course step through the first node of collecting user name to view the collect age node.

- 1 **Deploy** and try your changes.
- 2 Enter a first and last name, you can just simply enter one character.
- 3 Your collect age will look like the following screenshot.



Beginner's Luck Online Gaming

Verify Age

Hi [FNAME]!

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

Age

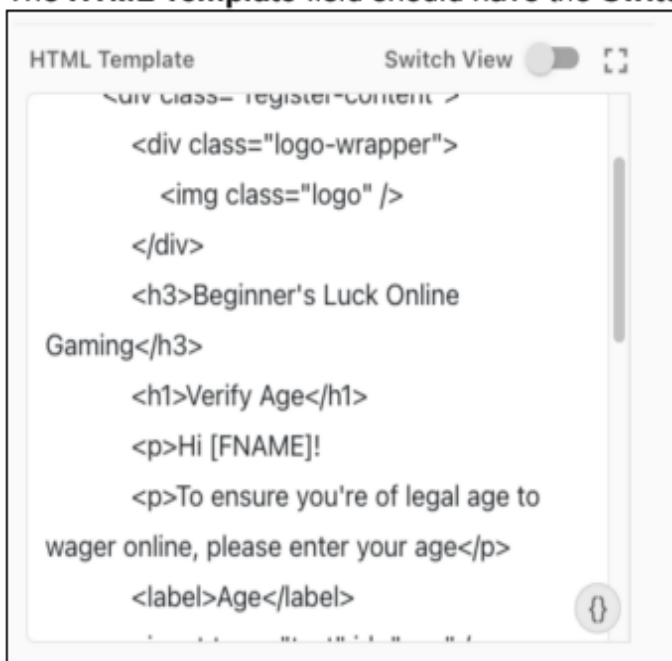
Next

- Note the focus on age and of course the previous page had the focus on first name.
- 4 Note the text [FNAME] on the form looks like there is a variable reference that needs to be fixed.
 - 5 Continue with the next section.

2.1.2.2 Fix the variable reference in template

The template has a variable reference to display the users first name and this was not set properly in the template.

- 1 In your flow click **Collect Age** node.
- 2 The **HTML Template** field should have the **Switch View** toggle off.



HTML Template Switch View ☐

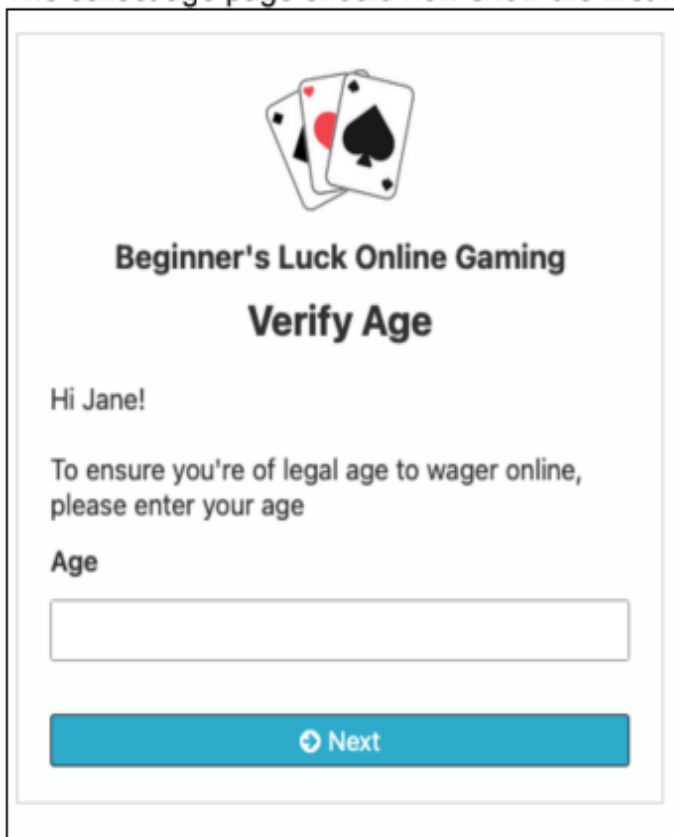
```
<div class="register-content">
  <div class="logo-wrapper">
    <img class="logo" />
  </div>
  <h3>Beginner's Luck Online
  Gaming</h3>
  <h1>Verify Age</h1>
  <p>Hi [FNAME]!
  <p>To ensure you're of legal age to
  wager online, please enter your age</p>
  <label>Age</label>
```

- 3 Highlight the text [FNAME] in the template.

- 4 Click  icon to select from the **First & Last Name** http node the **firstName** variable.



- 5 Apply your changes.
 6 Deploy, and try your flow again.
 7 The collect age page should now show the first name:



- 8 Continue with the next section.

2.1.2.3 Change age verification to use new variable

You now have a new node to collect the user age but the nodes that verify that the user is of legal age still have references to the old node variable. You need to fix these nodes in your flow.

- 1 Click the **Age Verification** node to edit it.
 - 1.1 In the **Value A** field remove the existing **age** variable.
 - You will notice that it has the icon for the screen node.
 - 1.2 Click the **{}** to select **Collect Age** node.
 - 1.3 Select the **age** variable.
 - 1.4 Apply and close the dialog.
- 2 Click the **Calculate Years** node to edit it.
 - 2.1 In the **age** input variable:
 - 2.1.1 In the **Value** field remove the existing **age** variable.
 - 2.2 Click the **{}** to select **Collect Age** node.
 - 2.3 Select the **age** variable.
 - 2.4 Apply and close the dialog.
- 3 You will test this in the next set of UI changes you make to the flow, because you will be executing past this block.
- 4 Continue with the next section.

2.1.3 Edit node Collect Email

This node is pretty much the same as the previous node therefore straight forward in the steps. The only difference is the password field you will provide some JavaScript that will allow the user to toggle the masking of the field. It improves the experience by giving the user the flexibility.

- 1 Add a new **Http** node to the canvas.
- 2 Click on the **Http** node the select the **Custom HTML Template** capability from the list.
 - 2.1 Next to **HTML Template** field toggle **Switch View** on to change to the code editor for the field.

2.2 Copy the following block of code and paste it into the **HTML Template** field.

```
<section class="register">
  <form id="frmRegister">
    <div class="register-content">
      <div class="logo-wrapper">
        <img class="logo" />
      </div>
      <h3>Beginner's Luck Online Gaming</h3>
      <h1>Email and Password</h1>
      <p>Almost done! Enter your email address
        and set a password</p>
      <label>Email</label>
      <input type="text" id="email" />
      <label>Password</label>
      <input type="password" id="password" /><i
        class="fa fa-eye togglepwd"
        id="togglepwd" aria-hidden="true"></i>
      <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>
```

2.3 Scroll down to the bottom to the **Output Fields List** to add fields needed for the form.

2.4 Click **Add** button to a new field.

2.4.1 Set the **Property Name** to **email**

- Case is important and must match what is in your template HTML.

2.4.2 The **Data Type** is **String**

2.4.3 You can take the defaults for everything else.

2.5 Click **Add** button to a new field.

2.5.1 Set the **Property Name** to **password**

- Case is important and must match what is in your template HTML.

2.5.2 The **Data Type** is **String**

2.5.3 You can take the defaults for everything else.

- The hiding and display of the field will be controlled via JavaScript and CSS

2.6 Copy the following block of code into the **Script** field.

```
document.getElementById("email").focus()

// Toggle password field visibility
document.getElementById("togglepwd").addEventListener('click', function (e) {
  let pwdField = document.getElementById("password");

  if (pwdField.type === 'password') {
    pwdField.type = 'text';
    e.currentTarget.classList.remove("fa-eye");
    e.currentTarget.classList.add("fa-eye-slash");
  } else {
    pwdField.type = 'password';
    e.currentTarget.classList.remove("fa-eye-slash");
    e.currentTarget.classList.add("fa-eye");
  }
});
```

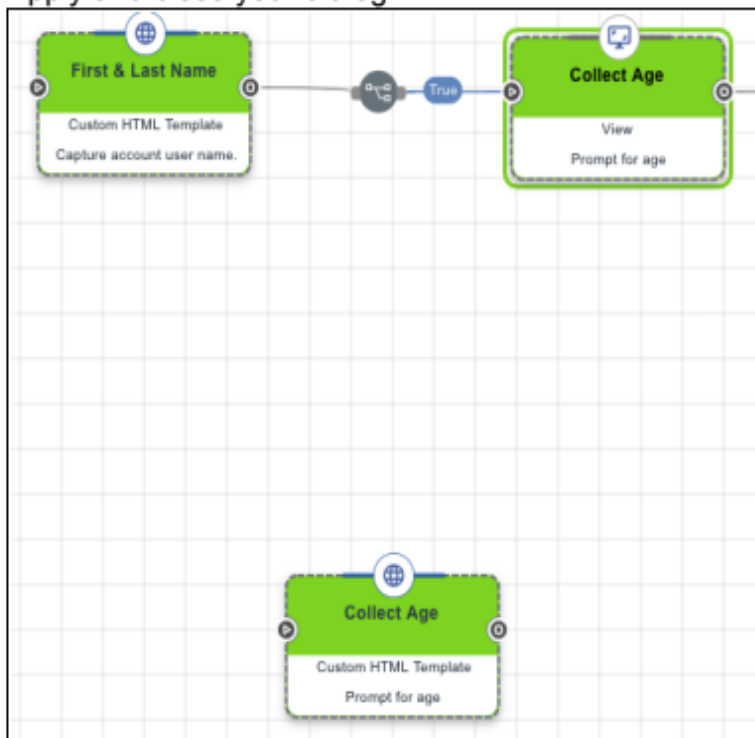
2.7 Click **Settings** tab at the top to name your node.

2.7.1 Set the **Node Title** to **Collect Email**

2.7.2 Set the **Node Description** to **Email & Password**

2.7.3 Set the **Node Background Color** to **lite green**

2.8 Apply and close your dialog.



3 Right-click on the node old screen node currently in the flow as **Collect Email** type node.

4 Select **Delete** menu item to delete the node.

5 Connect the new **Collect Age** node to hold the same spot in the flow as the old one.

6 Move the node up along the existing flow line to make your flow look neater.



7 Click **Save** button to save your flow.

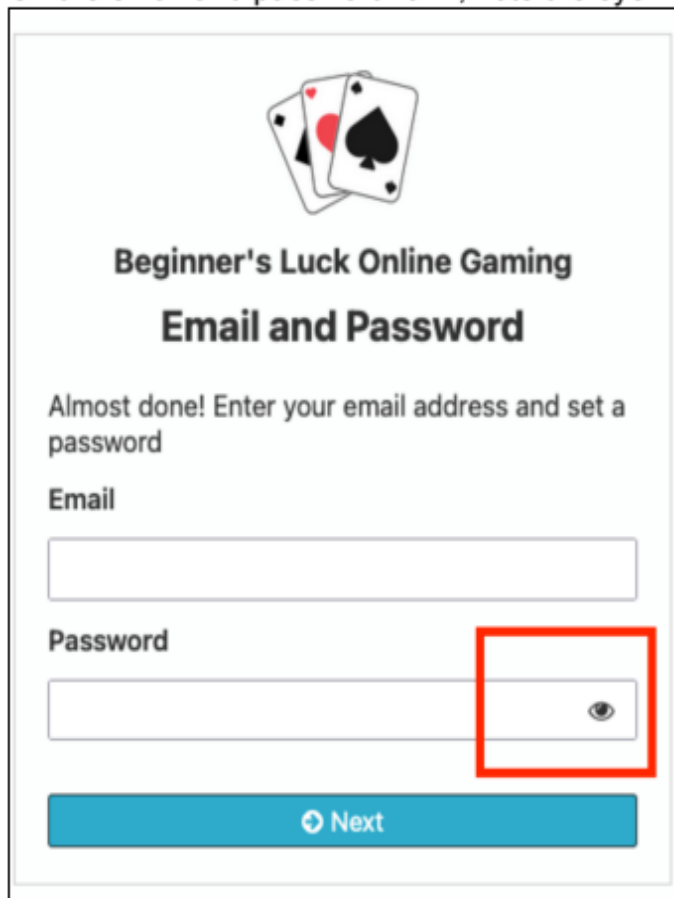
8 Continue with the next section.


2.1.3.1 Test changes to Collect Email node

As before you will need to step through the earlier pages. In this section you also want to test the toggle for password input field.

Getting through to this page will take you through the changes you made to the age verification part of the flow to use the new variable.

- 1 Deploy and try your changes.
- 2 Enter a first and last name, you can just simply enter one character.
- 3 Enter a valid age over 21.
- 4 On the email and password form, note the eye in the password field:






Beginner's Luck Online Gaming


Email and Password

Almost done! Enter your email address and set a password

Email


Password




 Next

- This is provided by JavaScript and CSS to provide a toggle to display or hide the password field.

5 Enter some values into the fields and try the toggle.

Password


- Toggle on.

Password


- Toggle off.

6 Continue with the next section.

2.1.4 Edit node Review Account Info

The account review page is a bit different than the other pages changed so far, it has no properties. The page is just simply used to display what was entered in earlier forms, so it did not need properties as it was displaying a custom message. But the template you are using does have input fields that reference properties therefore you will need to account for these properties.

- 1 Click **Review Account Info** node in your flow.

< **TRIGGER:**

Custom Html Message

GENERAL SETTINGS >

Message Title * indicates required

Verify New Account Details

Message Title CSS

Message

firstName, please review your new account details

First Name: firstName

Last Name: lastName

Age: age

Email: email

Password: password

- There are variable references but if you scroll down you will see no properties.
- 2 At the top of the dialog click < to the left of **TRIGGER** to see the list of capabilities for the node.
 - 3 Change the capability to **Custom HTML Template** for the node.

4 Copy the following block of code and paste it into the **HTML Template** field.

```
<section class="register">
  <div class="register-content">
    <div class="logo-wrapper">
      <img class="logo" />
    </div>
    <h3>Beginner's Luck Online Gaming</h3>
    <h1>Verify Account Details</h1>
    <div class="formfield-wrapper">
      <div>
        <label>First name</label>
        <input type="text" id="firstName" readonly />
      </div>
      <div>
        <label>Last name</label>
        <input type="text" id="lastName" readonly />
      </div>
    </div>
    <label>Email</label>
    <input type="text" id="email" readonly />
    <label>Password</label>
    <input type="password" id="password" readonly /><i class="fa fa-eye togglepwd"
      id="togglepwd"
      aria-hidden="true"></i>
    <div class="buttonbar">
      <button id="btnRegister" data-skcomponent="skbutton"
        data-skbuttontype="next-event"
        data-skbuttonvalue="register">
        <i class="fa fa-check-circle-o"
          aria-hidden="true"></i> Confirm
      </button>
    </div>
  </div>
</section>
```

5 Copy the following block of code into the **Script** field.

```
// Toggle password field visibility
document.getElementById("togglepwd").addEventListener('click', function (e) {
  let pwdField = document.getElementById("password");

  if (pwdField.type === 'password') {
    pwdField.type = 'text';
    e.currentTarget.classList.remove("fa-eye");
    e.currentTarget.classList.add("fa-eye-slash");
  } else {
    pwdField.type = 'password';
    e.currentTarget.classList.remove("fa-eye-slash");
    e.currentTarget.classList.add("fa-eye");
  }
});
```

- 6 Good idea at this time to Apply your changes to save what you have done so far, if you have not done it already.
- 7 In the **General** tab of the dialog, scroll down the **Output Fields List**
- 8 You need to all the fields and provide the values as variables as follows:
 - 8.1 Click **Add** button
 - 8.1.1 **Property Name** is
 - **firstName**
 - 8.1.2 **Value** select the variable **firstName** from the **First & Last Name** node.

8.1.3 For the remaining options for this field take the defaults.

The screenshot shows a configuration panel for a form field. It contains the following sections:

- Property Name:** A text input field containing the value "firstName".
- Display Name:** An empty text input field.
- Control Type:** A dropdown menu with "Text Field" selected.
- Data Type:** A dropdown menu with "String" selected.
- Secure:** A toggle switch that is currently turned off.
- Value:** A variable selection field showing "firstName" with a dropdown arrow on the right.

- The display name is handled by template and the data type is what is needed.

8.2 Click **Add** button8.2.1 **Property Name** is

- **lastName**

8.2.2 **Value** select the variable **lastName** from the **First & Last Name** node.

8.2.3 For the remaining options for this field take the defaults.

8.3 Click **Add** button8.3.1 **Property Name** is

- **age**

8.3.2 **Data Type** select **Number** from the dropdown.8.3.3 **Value** select the variable **age** from the **Collect Age** node.

8.3.4 For the remaining options for this field take the defaults.

8.4 Click **Add** button8.4.1 **Property Name** is

- **email**

8.4.2 **Value** select the variable **email** from the **Collect Email** node.

8.4.3 For the remaining options of this field take the defaults.

8.5 Click **Add** button8.5.1 **Property Name** is

- **password**

8.5.2 **Value** select the variable **password** from the **Collect Email** node.8.5.3 **Secure** toggle it on.

8.5.4 For the remaining options of this field take the defaults.

Property Name

password

Display Name

Control Type

Text Field ▾

Data Type

String ▾

Secure

☐

Value

password

9 Apply your changes and close the dialog.


10 Continue with the next section.

2.1.4.1 Test changes to Review Account Info node

As before you will need to step through the earlier pages. In this section you also want to test the toggle for password input field.

- 1 Deploy and try your changes.
- 2 Enter a first and last name, you can just simply enter one character.
- 3 Enter a valid age over 21.
- 4 Enter a email and password.

5 At the review form note that you can toggle the password field as before.



Beginner's Luck Online Gaming

Verify Account Details

First name

Sam


Last name

Second


Email

samsecond@example.com

Password

..... 

Confirm



Beginner's Luck Online Gaming

Verify Account Details

First name

Sam


Last name

Second

Email

samsecond@example.com

Password

2345Password@ 

Confirm

6 Continue with the next section.

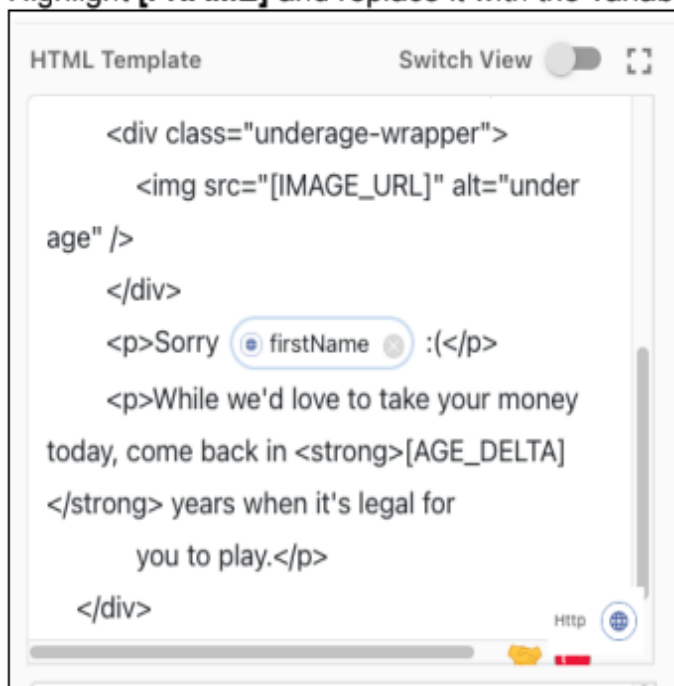
2.1.5 Edit node Not Legal Age

This particular form just simply displays information it does not have any fields, so no properties will be required.

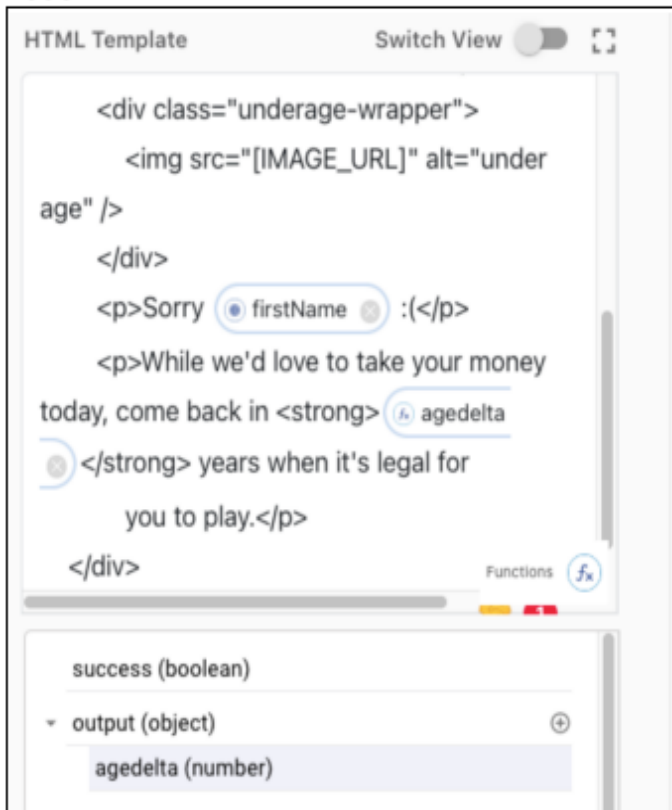
- 1 Click **Not Legal Age** node in your flow.
- 2 Change the capability to **Custom HTML Template** for the node.
- 3 **Copy** the following block of code and paste it into the **HTML Template** field.

```
<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">
      
    </div>
    <p>Sorry [FNAME] :(</p>
    <p>While we'd love to take your money today, come back
      in <strong>[AGE_DELTA]</strong>
      years when it's legal for
      you to play.</p>
    </div>
  </section>
```

- 4 There is no JavaScript needed for this template.
- 5 Look at the template you pasted and note the reference to **[IMAGE_URL]**, **[FNAME]**, and **[AGE_DELTA]** that are variables references to replace.
 - The use of **[]** and all uppercase is just a style selected to make it easier to indicate what is dynamic information on the form for display.
- 6 The reference for **[IMAGE_URL]** will be dealt with in a later exercise, so for now just leave as is.
- 7 Highlight **[FNAME]** and replace it with the variable **firstName** from the **First & Last Name** node.



- 8 Highlight **[AGE_DELTA]** and replace it with the variable **agedelta** from the **Calculate Years** node.

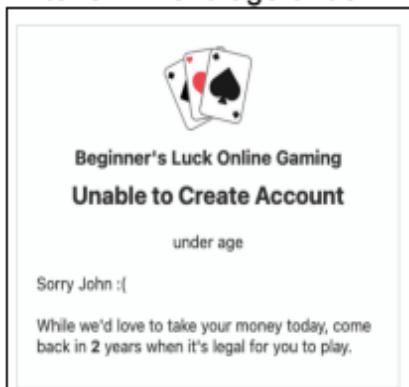


- 9 Apply your changes and close the dialog.
10 Continue with the next section.

2.1.5.1 Test changes to Not Legal Age node

Now it is time to test this message display and verify that it works.

- 1 Deploy and try your changes.
- 2 Enter a first and last name, you can just simply enter one character.
- 3 Enter an invalid age under 21.

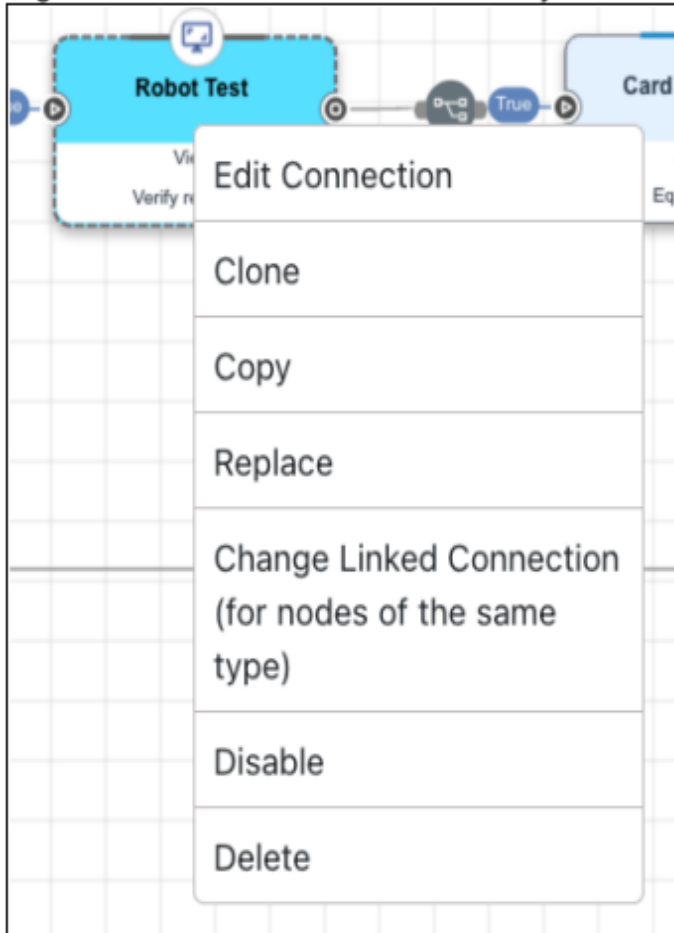


- 4 Continue with the next section.

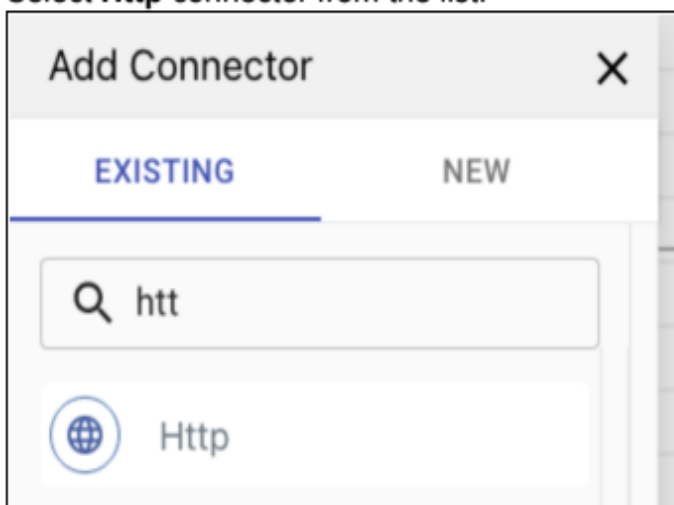
2.1.6 Edit node Robot Test

This node is not an HTTP connector but rather it is a Screen connector like some of the ones you updated earlier. But this time you will replace directly. You will also need to deal with the properties and variable references on the template.

- 1 Right-mouse click **Robot Test** node in your flow.



- 2 Click **Replace** option from the menu.
- 3 Select **Http** connector from the list.



- 4 Click on the **Http** node you just replaced on the canvas.
- 5 Select the capability of **Custom HTML Template** for the node.

6 Copy the following block of code and paste it into the **HTML Template** field.

```
<section class="register">
  <form id="frmRegister">
    <div class="register-content">
      <div class="logo-wrapper">
        <img class="logo" />
      </div>
      <h3>Beginner's Luck Online Gaming</h3>
      <h1>Human or Robot?</h1>
      <p>[FNAME], to prevent robots from winning all our money,
        please identify the playing card below</p>
      <div class="captcha-image">
        
      </div>
      <div class="formfield-wrapper">
        <div>
          <label>Face Value</label>
          <select id="cardvalue">
            <option value="0">Select Card Value</option>
            <option value="ACE">Ace</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
            <option value="8">8</option>
            <option value="9">9</option>
            <option value="10">10</option>
            <option value="JACK">Jack</option>
            <option value="QUEEN">Queen</option>
            <option value="KING">King</option>
          </select>
        </div>
        <div>
          <label>Card Suit</label>
          <select id="cardsuit">
            <option value="0">Select Card Suit</option>
            <option value="CLUBS">♣ Clubs</option>
            <option value="DIAMONDS">♦ Diamonds</option>
            <option value="HEARTS">♥ Hearts</option>
            <option value="SPADES">♠ Spades</option>
          </select>
        </div>
      </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>
```

7 There is no JavaScript needed for this template.

8 In the **General** tab of the dialog, scroll down the **Output Fields List**

9 You need to add two fields but for the values they will be input so you will not provide a value:

9.1 Click **Add** button

9.1.1 **Property Name** is

▪ **cardvalue**

9.1.2 For the remaining options for this field take the defaults.

Property Name
cardvalue

Display Name

Control Type
Text Field ▾

Data Type
String ▾

Secure
☐

Value

9.2 Click **Add** button9.2.1 **Property Name** is

- **cardsuit**

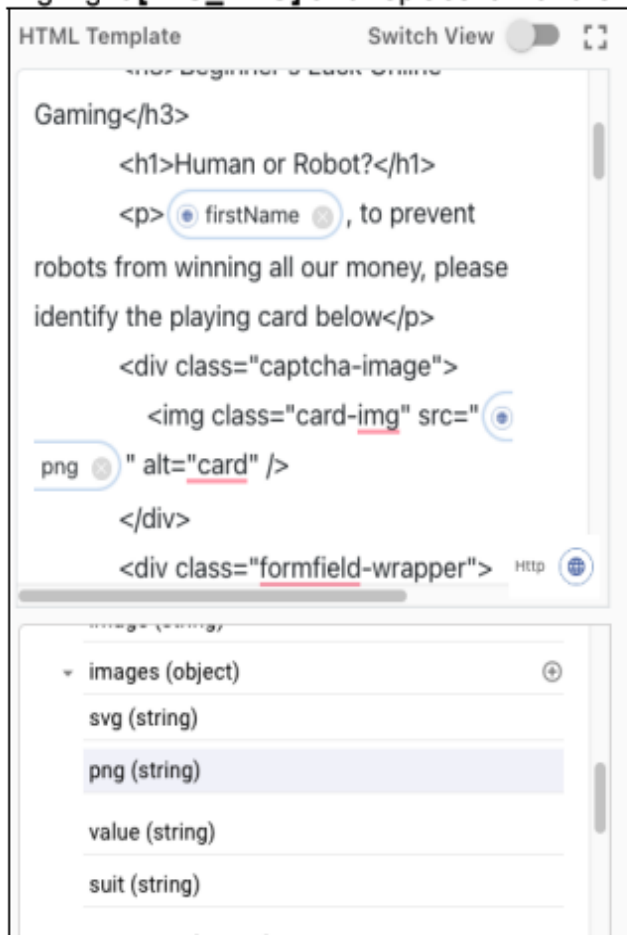
9.2.2 For the remaining options for this field take the defaults.

10 In the **HTML Template** field in non-code view.10.1 Highlight **[FNAME]** and replace it with the variable **firstName** from the **First & Last Name** node.

HTML Template Switch View ☐

```

Gaming</h3>
<h1>Human or Robot?</h1>
<p>• firstName , to prevent
robots from winning all our money, please
identify the playing card below</p>
<div class="captcha-image">
  
</div>
<div class="formfield-wrapper">
  
```

10.2 Highlight **[IMG_PNG]** and replace it with the variable **png** from the **Deck of Cards** node.

11 Apply your changes to this point to save it.

12 In the **Settings** tab.

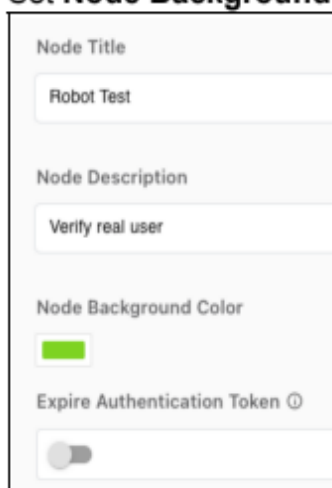
12.1 Set **Node Title** as

- **Robot Test**

12.2 Set **Node Descriptions** as

- **Verify real user**

12.3 Set **Node Background Color** as **lite green**.



13 Apply and close the dialog.



14 Continue with the next section.

2.1.6.1 Test changes to Robot Test node

Now it is time to test robot test display. The input selection and routing the next two displays will be done once you complete the remaining two user interaction pages to be updated.

- 1 Deploy and try your changes.
- 2 Enter a first and last name, you can just simply enter one character.
- 3 Enter a valid age over 21.
- 4 Enter a email and password.
- 5 Continue through the account review.
- 6 A random card will of course display and you will have two dropdown fields to allow the user to select a value.



Beginner's Luck Online Gaming

Human or Robot?

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



Face Value

Select Card Value ▾

Card Suit

Select Card Suit ▾

Next

- 7 Continue with the next section.

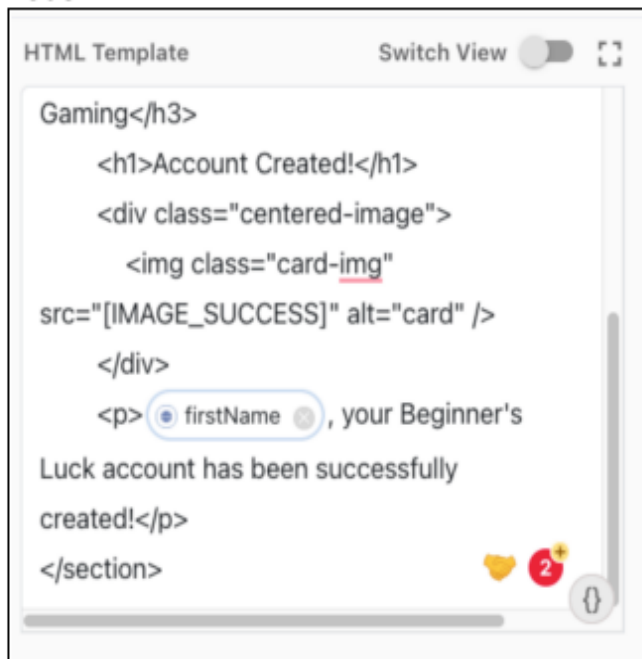
2.1.7 Edit node Account Created

This next page just simply displays a message to the user and has no input, so it is pretty straight forward and requires no property additions.

- 1 Click **Account Created** node in your flow.
- 2 Change the capability to **Custom HTML Template** for the node.
- 3 **Copy** the following block of code and paste it into the **HTML Template** field.

```
<section class="register">
  <div class="register-content">
    <div class="logo-wrapper">
      <img class="logo" />
    </div>
    <h3>Beginner's Luck Online Gaming</h3>
    <h1>Account Created!</h1>
    <div class="centered-image">
      
    </div>
    <p>[FNAME], your Beginner's Luck account has been successfully created!</p>
  </div>
</section>
```

- 4 There is no JavaScript needed for this template.
- 5 In the **HTML Template** field in non-code view.
 - 5.1 Highlight **[FNAME]** and replace it with the variable **firstName** from the **First & Last Name** node.



- 5.2 The **[IMAGE_SUCCESS]** will be replaced in a later exercise.
- 6 Apply your changes to this point to save it.
 - You will test these changes along with the changes in the next section together.
- 7 Continue with the next section.

2.1.8 Edit node Robot Test Failed

This is the last page to be changed and again just simply displays some text with no input fields.

- 1 Click **Robot Test Failed** node in your flow.
- 2 Change the capability to **Custom HTML Template** for the node.
- 3 **Copy** the following block of code and paste it into the **HTML Template** field.

```
<section class="register">
  <div class="register-content">
    <div class="logo-wrapper">
      <img class="logo" />
    </div>
    <h3>Beginner's Luck Online Gaming</h3>
    <h1>Robot Detected!</h1>
    <p>[FNAME], we think you are a robot. If you promise you're not, try again.
    </p>
    <div class="centered-image">
      
    </div>
    <div class="buttonbar">
      <button id="btnRegister" data-skcomponent="skbutton"
        data-skbuttontype="next-event"
        data-skbuttonvalue="register">
        <i class="fa fa-repeat" aria-hidden="true"></i> Retry
      </button>
    </div>
  </div>
</section>
```


- 4 There is no JavaScript needed for this template.
- 5 In the **HTML Template** field in non-code view.
 - 5.1 Highlight **[FNAME]** and replace it with the variable **firstName** from the **First & Last Name** node.
 - 5.2 The **[IMG_ROBOT_DECTED]** will be replaced in a later exercise.
- 6 Apply your changes to this point to save it.
- 7 Continue with the next section.

2.2 Test the complete flow changes

In this section you will test the complete flow where you will deal with the last two pages that were changed. The robot test failure and creation of the account.

- 1 Save, deploy, and try your changes.
- 2 Enter a first and last name, you can just simply enter one character.
- 3 Enter a valid age over 21.
- 4 Enter a email and password.
- 5 Continue through the account review.


6 At the robot test select an invalid suit or value for the card.



Beginner's Luck Online Gaming

Human or Robot?

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



Face Value


2

Card Suit

♠ Spades

Next

7 Your robot test page will display:



Beginner's Luck Online Gaming

Robot Detected!

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

robot detected

Retry

8 Click **Retry** button.

9 At the robot test select an valid suit and value for the card.



Beginner's Luck Online Gaming

Human or Robot?

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



Face Value


8

Card Suit

♥ Hearts

Next

10 The account created screen will display:



Beginner's Luck Online Gaming

Account Created!

card

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

11 You have completed this exercise.