

Create User Account when Registration Complete

Table of Contents

- 1 Objective
- 2 Do the following task items
 - 2.1 Review email service options in PingOne
 - 2.2 Initial review of PingOne setup
 - 2.2.1 Review the PingOne SSO connector
 - 2.3 Registration flow with PingOne integration
 - 2.3.1 Add PingOne node to flow
 - 2.3.2 Create a new flow policy in DaVinci application
 - 2.3.3 Update web application and test the new flow
 - 2.3.4 Review flow analytics and PingOne identities
 - 2.4 Error handling for failure to create user
 - 2.5 Check for existing user
 - 2.6 Email and password validation
 - 2.7 Email notification to created user

1 Objective

The main objective of this exercise is to take the information collected during the registration process and create a user account in PingOne. This is the first step to expanding the capabilities of your application to support authentication.

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.

Tip

You will be using Glitch for editing the code and testing with the your flow changes. Before you start a lab or need to access Glitch, ensure that you are logged into your Glitch account and accessing the current project.

2 Do the following task items

You will create a PingOne user if the user is of legal age and in the process do additional validation of the data entered. The flow will complete by emailing the user when the account is successfully created in PingOne.

Tip

You will now be creating user accounts in PingOne and in later exercises use these accounts to authenticate. You will be defining the passwords for these accounts to comply with PingOne password policies. Make note of these passwords and emails for the accounts, so that you can use them later during the authentication flow development and testing.

2.1 Review email service options in PingOne

Depending on the PingOne license you are using for this training you may not be able to use the PingOne email service to send email notifications. This section will take you through the process of reviewing your PingOne setup and dealing with the configuration to send emails.

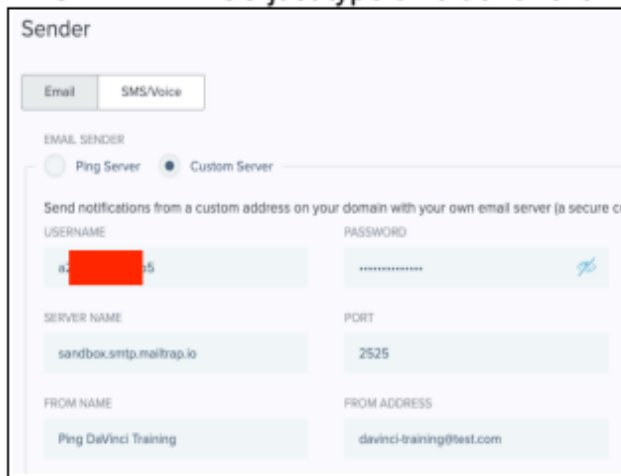
It is this section that will determine if you definitely need a free [Mailtrap](#) account. Or if it will be optional depending how you want to deal with emails that will be sent and you need to respond to. In the prior lab if you successfully sent the email then you don't need the Mailtrap account, but you may want to use it or something else (example [Mailinator](#)) to deal with test emails. In future exercises you will need to deal with emails for the test users you will be creating and they will need unique email addresses. Another option if your email account allows for alias accounts of course you can use that for your test accounts.

- 1 In a browser tab **open** your **PingOne** administrator console.
- 2 You should be in **Intro DaVinci Training** environment.
- 3 Click on **User Experience** in the left menu to open it.
- 4 Click on **Notification Templates** in the left menu.
- 5 Look at the **Notification Templates** title on the main part of the page, if you see the **blue +** icon next to it (see screenshot below) then you do not need to continue with this section, please continue on to the next section.



- 6 If you don't have the **+** icon or it is grayed out continue with the following steps.
 - 6.1 You will need to register for a free [Mailtrap](#) account to configure the following.
 - 6.2 In Mailtrap you will need to **Email Testing** then **Inboxes** opening the inbox you want to use, note this is typically **MyInbox**.
 - 6.3 You will need to go into **SMTP Settings** and click **Show Credentials** to expose the information you will need to configure PingOne.
 - 6.4 In PingOne console select **Sender** you will find it under **Settings** in the left menu.
 - 6.5 In the **Email** tab select the **Custom Server** radio button.
 - 6.6 Using the information from Mailtrap set the following values, first name is the PingOne field value and second name is the Mailtrap configuration name.

- 6.6.1 **USERNAME** value of **Username**
- 6.6.2 **PASSWORD** value of **Password**
- 6.6.3 **SERVER NAME** value of **Host**
- 6.6.4 **PORT** value of **Port** suggest that you use 2525
- 6.6.5 **FROM NAME** just type a value for example *Ping DaVinci Training*
- 6.6.6 **FROM ADDRESS** just type a value for example *davinci-training@test.com*



- 6.6.7 Click the **Send Test Email** button to send a test email to mailtrap.
- 6.6.8 You will see that email in Mailtrap inbox.
- 6.7 Click **Save** button to save the changes in PingOne.
- 7 If you are using Mailtrap in this way then any email addresses you use in future exercises will go to this inbox no matter what email address you use for the user.
- 8 If you are not using Mailtrap then the email addresses you use for testing will need to be valid emails that will go to an inbox you have access to.
- 9 Continue with the next section.

2.2 Initial review of PingOne setup

The first thing that needs to be done is to create a population where your users will be placed in PingOne. You will also review the configuration of the PingOne SSO connector that you will be using to manage users in PingOne.

- 1 In a browser tab **open** your **PingOne** administrator console.
- 2 You should be in **Intro DaVinci Training** environment.
- 3 Click **Directory** from the left menu.
- 4 Click **Populations** from the left menu.
- 5 Click the **+** icon to create a population.
- 6 For the **Name** type
 - **Beginners Luck Users**

- 7 Accept the default for password policy and click **Save** button to create the population.



- 8 You can close the dialog to the right by clicking the **x**.
9 Continue with the next section.

2.2.1 Review the PingOne SSO connector

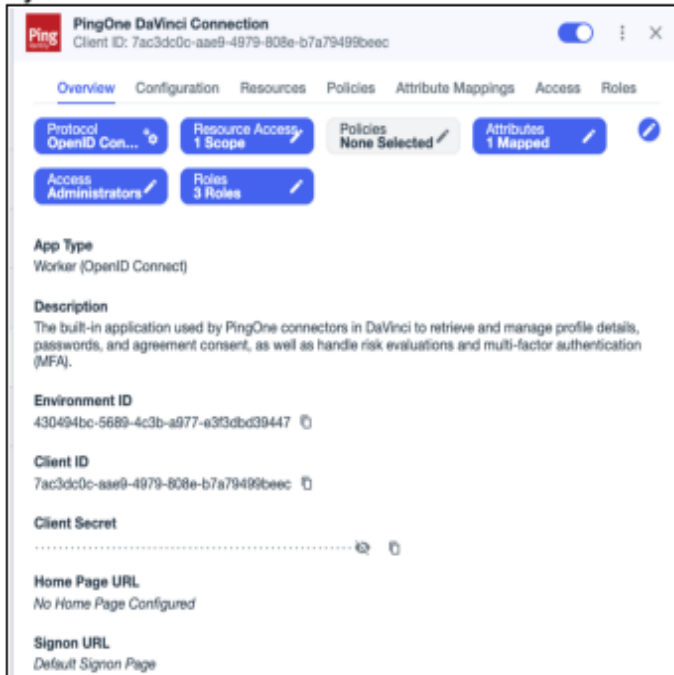
The connector you will use to create/manage users in PingOne is already defined in DaVinci. This was created when your PingOne DaVinci service was defined to provide the ability to SSO into DaVinci from PingOne. This connector can be used because it is configured to use the same tenant and environment where you want your user population to sit.

- 1 In the **DaVinci** administrator console.
- 2 Click **Connectors** in the left menu.
- 3 You will see a **PingOne** connector among other connectors, some of which you created in earlier exercises.
 - You can use the search field, reducing it down to just what you want to see.
- 4 You can use the **Add Connector** button to open up a dialog and search for other PingOne type connectors.
- 5 No need to create another one, select the **PingOne** that already exists to review the configuration.



- 6 The configuration shows what **environment** is being used by this connector. Generally you would only need a different connector if you were creating/managing in another environment.

- 7 The **client id** is the application in PingOne.
 - This was what you used in the earlier exercise when you setup the notifications connector.
- 8 In **PingOne** console select **Applications** then **Applications**
- 9 Review the configuration for **PingOne DaVinci Connection**, the worker application that is used by the connector in DaVinci.



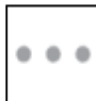
- 10 Continue with the next section.

2.3 Registration flow with PingOne integration

In this exercise you will build out a new flow use the existing flow as a starting base. The functionality for testing for robots will be removed because in a later exercise you will require verification of the email address, which will serve as a way to deal with robots.

Once you remove the nodes that are no longer needed you will add a PingOne node to create the user account in the next section.

- 1 In the **DaVinci** console select the **Flows** from the left menu.



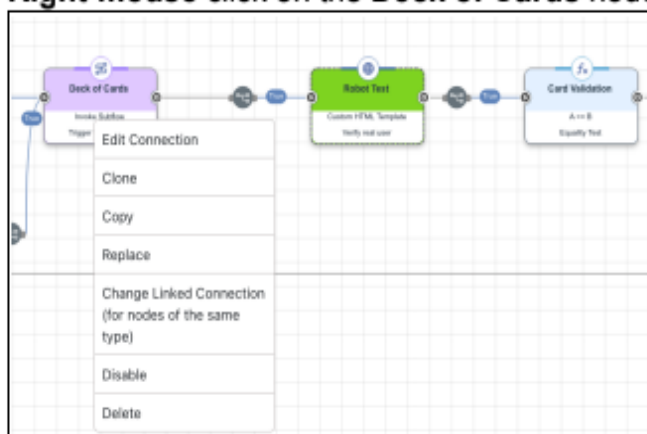
- 2 Click on the icon on the **Progressive Registration** flow.
- 3 Click **Clone** to copy the flow.
- 4 **Rename** the flow you just cloned as:
 - **Progressive Registration - PingOne User**



- 5 **Open** the new flow to edit it.
- 6 **Scroll** to the right in your flow until you see the **Deck of Cards** node.

7 You need to delete some nodes from your flow:

7.1 **Right-mouse** click on the **Deck of Cards** node and delete it.



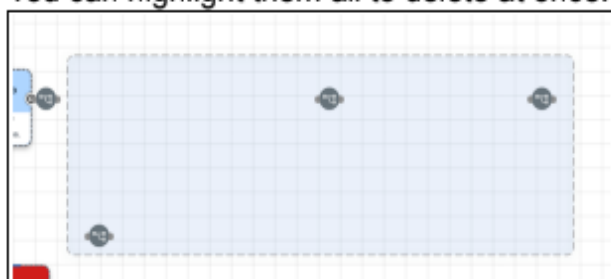
7.2 **Delete** the **Robot Test** node.

7.3 **Delete** the **Card Validation** node.

7.4 **Delete** the **Robot Test Failed** node.

7.5 **Delete** the **Action Decision** nodes left hanging.

- You can highlight them all to delete at once.



7.6 **Delete** the **Annotation** node at the top of this block.

8 Click **Save** button to save your changes to this point.

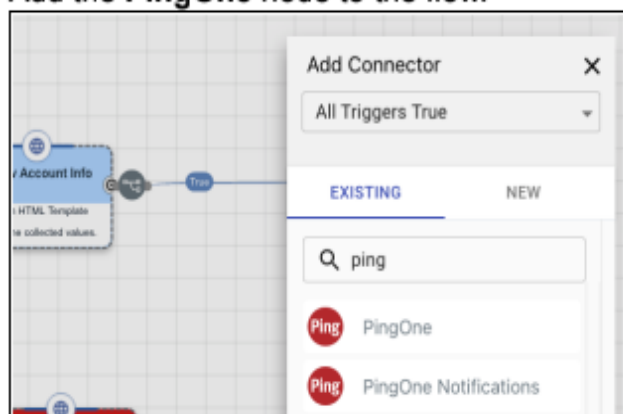
9 Continue with the next section.

2.3.1 Add PingOne node to flow

You have your new flow and you removed what you no longer need from it. Now it is time to setup the creation of a user in PingOne once the user successfully gets to the end of this flow.

1 At the **Action Decision** node after **Review Account Info** node draw a new path.

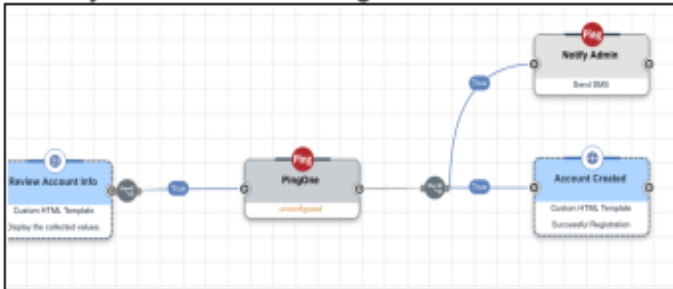
2 Add the **PingOne** node to the flow.



- 3 Connect the **PingOne** node to the **Action Decision** node that is before the **Account Created** node.



- 4 Move your nodes to the right so that there isn't as much space:



- 5 Click the **PingOne** node you just placed on the canvas.
- 5.1 Select the **Create User** capability
 - 5.2 In the **Population** dropdown select **Beginners Luck Users** from the list.
 - 5.3 In the **Username** field select the **email** variable from the **Collect Email** node.
 - You will use the email as the unique username that users will use to login to the application after they register.

Username * ⓘ

HTTP ⓘ

success (boolean)

▼ output (object) ⓘ

email (string)

password (string)

buttonValue (string)

▼ error (object) ⓘ

- 5.4 In the **Password** field select the **password** variable from the **Collect Email** node.
- 5.5 In the **Given Name** field select the **firstName** variable from the **First & Last Name** node.
- 5.6 In the **Family Name** field select the **lastName** variable from the **First & Last Name** node.

5.7 In the **Email** field select the **email** variable from the **Collect Email** node.

The screenshot shows a registration form with the following fields and their selected variables:

- Username**: email
- Population**: Beginners Luck Users
- Password**: password
- Given Name**: firstName
- Family Name**: lastName
- Email**: email

5.8 Click **Apply** button to save your changes to this point.

6 **Scroll** down until you see the **Other Attributes** at the bottom of attributes listed.

- You have the country value that was passed into the flow and will save this with the address attribute in PingOne.

7 Click **Add** button to add a new attribute to the list.

7.1 In the **Variable Name** field use the dropdown to select from the very long list.

7.1.1 Select **address** as the attribute.

The screenshot shows a dropdown menu for the 'Variable Name' field. The menu is open, displaying a list of attributes. The attribute 'address' is highlighted with a red rectangular box. Other visible attributes include 'Select..', 'account', 'accountId', 'createdAt', 'enabled', 'externalId', and 'id'.

- This attribute is of type object and the value you want is actually contained in this attribute.

- The object is a JSON object and you will use an expression to update the country part of the address

7.2 Select **Data Type** as **Object** from the dropdown list.

7.3 In the **Value** field type the `{"countryCode": ""}` expression.

7.3.1 In the second set of "" select the **Global** variable parameter **countryCode** that is passed into the flow by the web application.



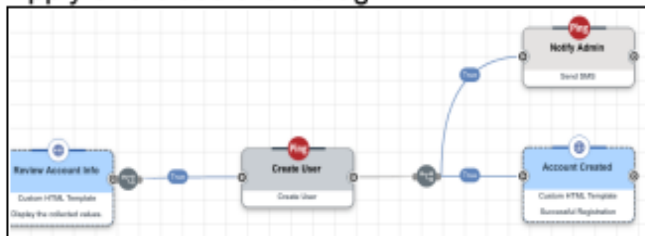
7.4 Click **Apply** button to save your changes to this point.

- You did not save the age value as there is no attribute in PingOne for this value. You could extend the PingOne directory schema to include this attribute and then add it to the user object. This will be left as an exercise if you have time or for after this training.

8 Click **Settings** tab.

8.1 For **Node Title** field type **Create User**

8.2 Apply and close the dialog.



9 Deploy your changes to the flow.

10 Continue with the next section.

2.3.2 Create a new flow policy in DaVinci application

Instead of updating the existing flow policy in the application to point to your new flow, you will create a new flow policy. This of course will require a change to the web application to use the new policy. You could change the policy in DaVinci instead and not have to change the web application, this is more of a process decision that you make in your organization to how you want to treat flow changes.

- 1 In **DaVinci** select **Applications** in the left menu.
- 2 Click **Beginners Luck App** to edit it.
- 3 Click **Flow Policy** tab to list out existing policies.

4 **Add** a new policy called **Registration PingOne**

4.1 Select the **Progressive Registration - PingOne User** flow from the list.

4.2 Check the **Latest Version** checkbox for the policy.

4.3 Create the new policy with defaults on the weight distribution.

- 5 **Copy** the policy ID to update your web application in the next section.
- 6 Continue with the next section.

2.3.3 Update web application and test the new flow

Now that your flow is ready and you have a policy ID it is time to update the web application to invoke this new policy and create the PingOne user account.

You may want to leave the browser tab with application open because you will be coming back to it for further testing.

- 1 In your **Glitch** application select the **index.html** file to edit it.
- 2 Find the function definition for `loadRegistrationFlow`
- 3 **Comment** out the existing `policyId=` line in the function.
- 4 **Add** a new line after the existing commented out line for `policyId`

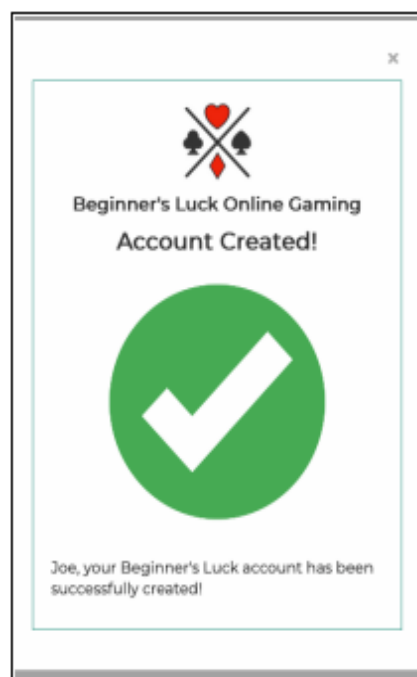
```
◦ const policyId=""
```

- 5 **Paste** the policy ID you copied in the previous section, inside the **quotes**

```
57 // Load DaVinci Registration Flow
58 function loadRegistrationFlow() {
59   // const policyId = "28c[REDACTED]e3"
60   // const policyId = "d7a[REDACTED]09"
61   const policyId = "25a[REDACTED]e49"
62   const divComponent = "flow-widget"
63   let flowInputData = {"countryCode": COUNTRY_CODE}
64
65   loadwidget(policyId, divComponent, flowInputData)
66 }
```

- 6 Click the **Preview** button at the bottom of the page and then select **Preview in a new window**
- 7 Go to the new tab where your application is now running.
- 8 Click the **Register** button and enter a new account, going through the end of the process

- 8.1 The password should be 8 characters or more with a mix of letters (upper/lower), numbers, and special characters to be accepted.
- You will be dealing with validation of passwords in a future section of this exercise.

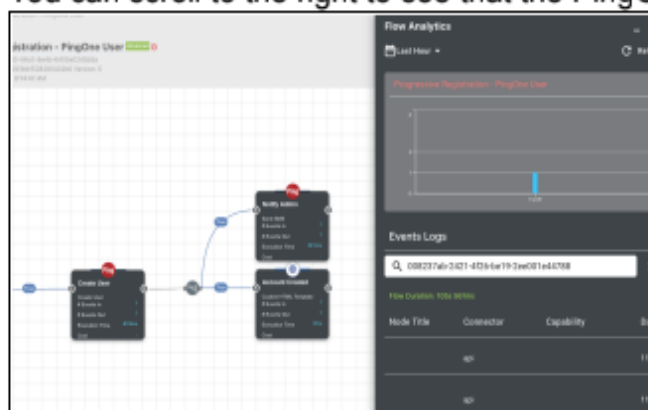


9 Continue with the next section.

2.3.4 Review flow analytics and PingOne identities

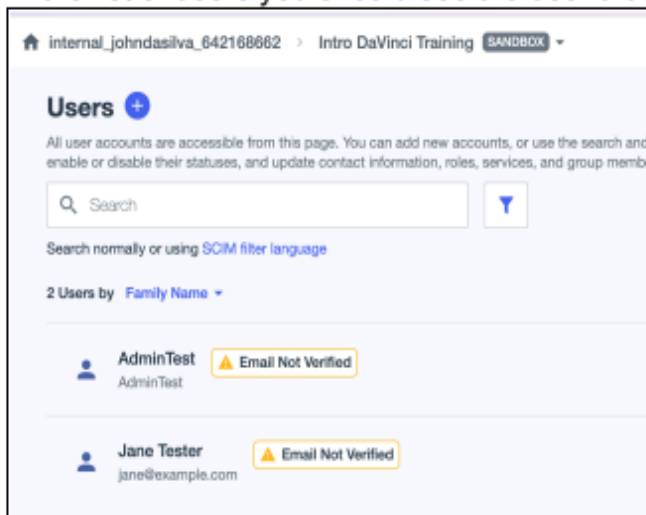
If everything worked as expected in the previous section you will have a user in your PingOne population.

- 1 In the **DaVinci** console select your **Progressive Registration - PingOne User** flow.
 - 1.1 Click **Analytics** button at the bottom of the canvas.
 - 1.2 Select the most recent **event** in the list.
 - 1.3 You can scroll to the right to see that the PingOne node was invoked.



- 1.4 Once you are done viewing the analytics you can click the **x** to close the dialog.
- 2 In the **PingOne** console select **Directory** then **Users** from the left menu.

2.1 In the list of users you should see the user that you just created.



- Of course your user details will differ but the key point is that the user is there.

2.2 Select **Populations** there you will see two populations with each having 1 user each.

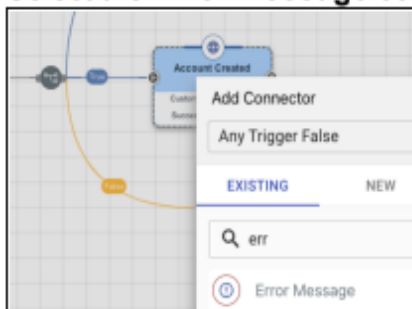
- The default one is the one you used in the an earlier exercise to set up a user to send email messages.
- You will be removing the email to administrator message and replace it with an email to the user created in a later section of this exercise.

3 Continue with the next section.

2.4 Error handling for failure to create user

You should anticipate that not all calls to Create User will be successful. Whether that's due to missing information or policy issues, let's build out a false branch to handle the unexpected.

- 1 In the **DaVinci** console select your **Progressive Registration - PingOne User** flow.
- 2 From the **Action Decision** node after **Create User** node drag to add a new node.
- 3 It should be set to **Any Trigger False** and search for **err**
- 4 Select the **Error Message** connector.



- 5 Click the **Error Message** node you just placed on the canvas to edit it.
- 6 Click **Custom Error Message on Screen** capability.

- 7 In the **Error Message** field select the variable **message** from the **error** response of the **Create User** node.

Error Message ⓘ

message ⓘ

PingOne **Ping**

success (boolean)

▸ output (object) ⓘ

▾ error (object) ⓘ

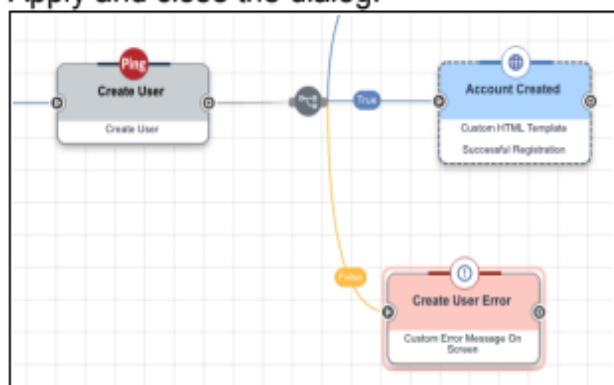
code (number, string)

message (string)

details (array)

8 Select **Settings** and set **Node Title** to **Create User Error**

9 Apply and close the dialog.



10 If needed click **Save** to save your changes to the flow to this point.

- You still need to deal with where the message is going to be displayed.
- This will be the user interaction node before Create User node which is Review Account Info node.

11 Click the **Review Account Info** node to edit the HTML template.

11.1 In the **HTML Template** you will be adding some HTML, this best by expanding the field to full screen for editing.

```

HTML: <script>
var d=document.getElementsByTagName("script");
var s=document.createElement("script");
s.src="//www.facebook.com/js/sdk_loader.js";
s.async=true;
s.id="facebook-jssdk";
d[0].parentNode.insertBefore(s,d[0]);
</script>

```

11.2 You will add your code after the **div class="buttonbar"** tag in the file.

11.3 **Copy** and paste the following code into a new line in the template.

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

```
<div class="buttonbar">
  <button id="btnRegister" data-skcomponents="skbutton" data-skbuttontype="next-event"
    data-skbuttonvalue="register">
    <i class="fa fa-check-circle-o" aria-hidden="true"></i> Confirm
  </button>
</div>
<div data-skcomponent="skerror" class="error-message" data-skvisibility="" ></div>
</div>
</section>
```

11.4 Apply and close the dialog to save your changes to the node.

12 Save and deploy this node in preparation for testing.

13 In the browser tab with your application, click **Register** button to create a new account and set the **password** to **123456**

Beginner's Luck Online Gaming
Verify Account Details

First name Last name
Fail Test

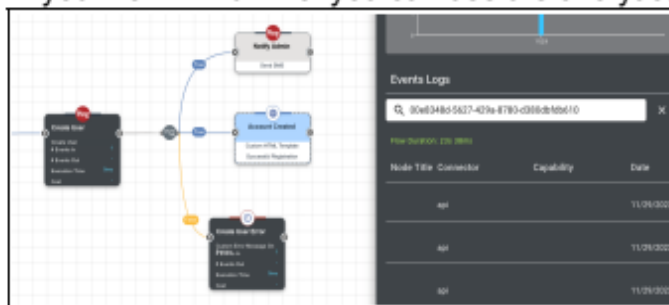
Email
john@example.com

Password

Confirm

password: User password did not satisfy password policy requirements

14 In your flow in **DaVinci** you can use the analytics to see the path taken.

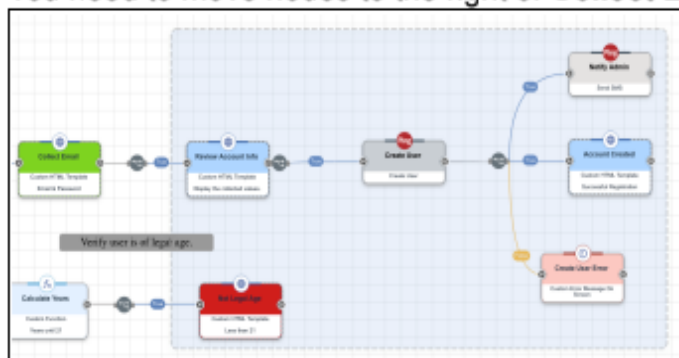


15 Continue with the next section.

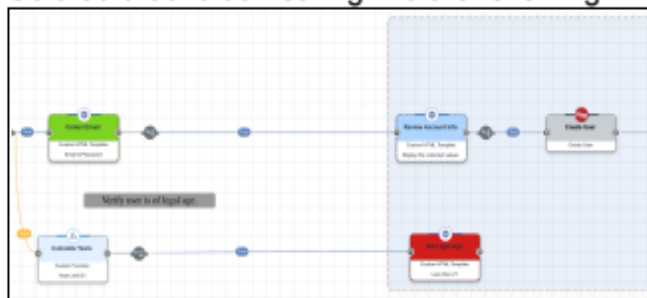
2.5 Check for existing user

Within your registration flow, you'll want to provide a rich user experience that provides early feedback to the user if the information they provide is incorrect or missing. Keeping this in mind, before creating a user, you should check to see if an account with their email already exists.

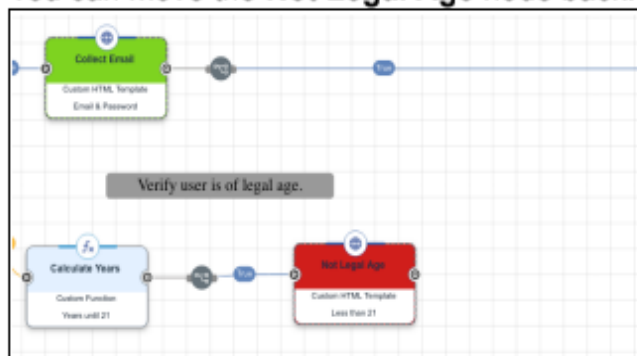
- 1 In your flow in **DaVinci** scroll over so you see your **Collect Email** node.
- 2 You need to move nodes to the right of **Collect Email** to make some space to add a new node.



2.1 So that it looks something like the following:



2.1.1 You can move the **Not Legal Age** node back.

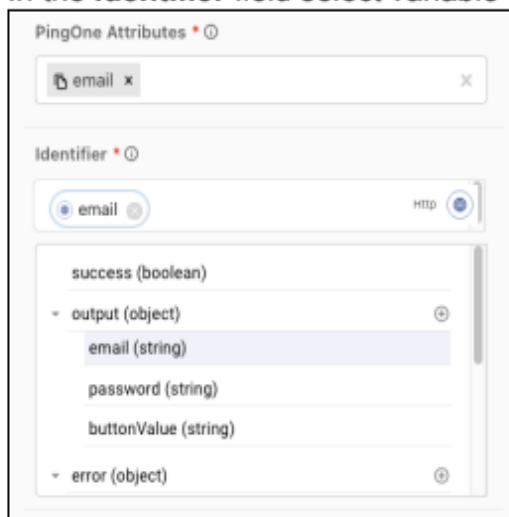


- 3 The line between the **Action Decision** node to the left of **Review Account Info** node, **delete** it.
- 4 Then **draw** a new line and add a **PingOne** node.
- 5 Connect the **PingOne** node to the **Review Account Info** node.



- 6 Click the **PingOne** node you just placed on the canvas to edit it.
 - 6.1 Click the **Find User** capability.
 - 6.2 In the **PingOne Attributes** field type **email** and press **return** key.

6.3 In the **Identifier** field select variable **email** from the **Collect Email** node.



6.4 Select **Settings** and set **Node Title** to **Find Existing User**

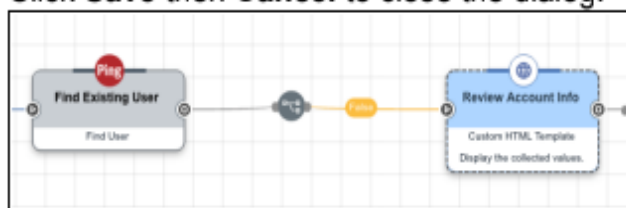
6.5 Apply and close the dialog.

7 Select the **Action Decision** node in front of **Find Existing User** node.

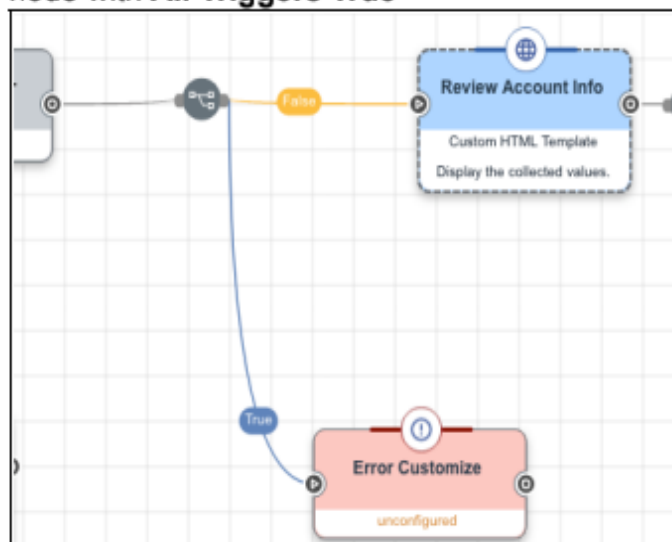
8 Change it to **Any Trigger False**

- If the user does not exist then we are good to continue otherwise you have an error.
- The error will end up on the true path that you will do next.

9 Click **Save** then **Cancel** to close the dialog.



10 From the **Action Decision** node you just edited, draw a line down and add a **Error Message** node with **All Triggers True**



11 Click the **Error Message** node to edit it.

11.1 Click the **Custom Error Message on Screen** capability.

11.2 In the **Error Message** field type or copy/paste the following:

- **An account already exists with this email address**

11.3 Select **Settings** and set **Node Title** to **Existing User Found**

11.4 Apply and close the dialog.

12 Click the **Collect Email** node to edit the HTML template.

12.1 In the **HTML Template** you will be adding some HTML, this best by expanding the field to

full screen for editing.

12.2 You will add your code after the **div class="buttonbar"** tag in the file.

12.3 **Copy** and paste the following code into a new line in the template.

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

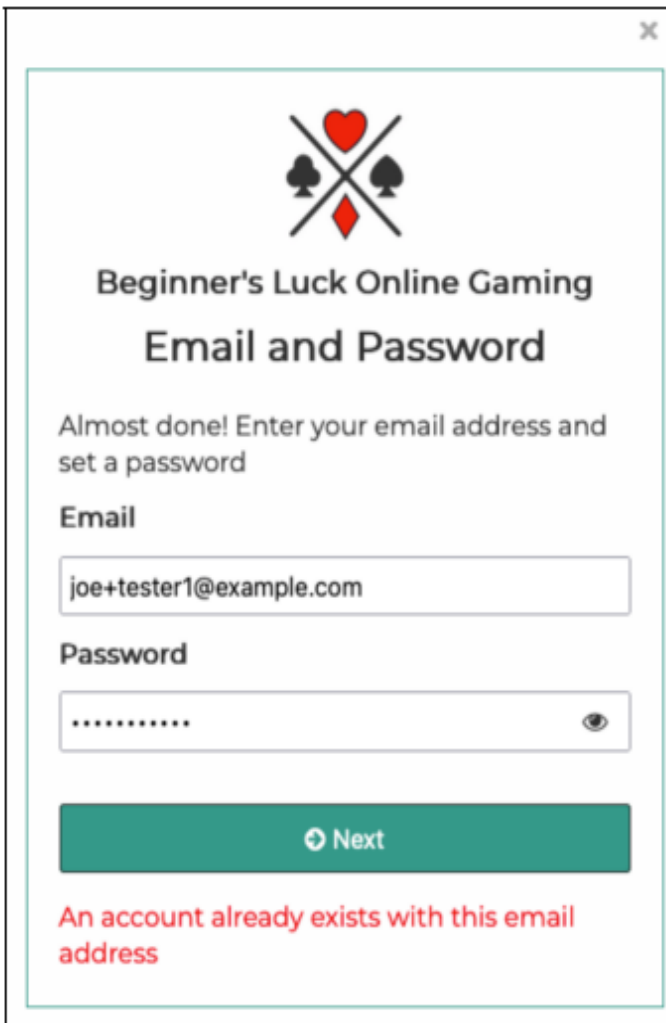
```
<div class="buttonbar">
  <button id="btnRegister" data-skcomponent="skbutton" data-skbuttontype="form-submit"
    data-skbuttonvalue="register" data-skform="frmRegister">
    <i class="fa fa-arrow-circle-right" aria-hidden="true"></i> Next
  </button>
</div>
<div data-skcomponent="skerror" class="error-message"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

12.4 Apply and close the dialog to save your changes to the node.

13 Save and deploy this node in preparation for testing.

14 In the browser tab with your application, click **Register** button, use the same email as the account you successfully created earlier.

- If you need to, go back into PingOne console to find that user and email.




Beginner's Luck Online Gaming

Email and Password

Almost done! Enter your email address and set a password

Email

Password

 Next

An account already exists with this email address

15 Continue with the next section.

2.6 Email and password validation

As a valid email address, and a sufficiently strong password are requirements when creating a PingOne account, let's update your HTML form to include field validation.

- 1 In the **Progressive Registration - PingOne User** flow.
- 2 Click the **Collect Email** node to edit it.
- 3 **Scroll** down to the **Form validation rules** section of the configuration.
- 4 Click **Add** button to define a rule.
 - 4.1 For **Property Name** type **email**
 - 4.2 Under **Rules** click **Add** to add a new rule.
 - 4.3 Select **Email** as the **Rule Name** from the dropdown list.
 - 4.4 In the **Validation Message** field type or copy/paste
 - **Please supply a valid email address**

Form validation rules ⓘ

Property Name ⊖

email

Rules

Rule Name Validation Message

Email ⊖ email Please supply a valid email address

ADD

- 5 Click **Add** button to define a rule.
 - 5.1 For **Property Name** type **password**
 - 5.2 Under **Rules** click **Add** to add a new rule.
 - 5.3 Select **Format** as the **Rule Name** from the dropdown list.
 - 5.4 In the **Regex** field copy and paste the following expression:

```
^(?=.*[0-9])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{8,24}$
```

- This expression ensures there is at least one number, one letter lower case, one letter upper case, one special character and the length is between 8 and 24 characters
- This is the default password policy in PingOne
- You can of course define your own policy but you would need to generate a proper regex for validating it

5.5 In the **Validation Message** field type or copy/paste

- **Password must contain at least one number, one letter lower case, one letter upper case, and a special character and be between 8 and 24 characters in length**

password

Rules

Rule Name

Format

Regex

`^(?=.*[0-9])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^`

Validation Message

password Password must contain at least on

6 In the **HTML Template** you will be adding some HTML, this is best done by expanding the field to full screen for editing.

6.1 You will add your code after the **div data-skcomponent="skerror"** tag in the file, which handles the existing user message.

- You will be adding two lines to handle the two new validation rules you added to the node.

6.2 **Copy** and paste the following two code lines in the template.

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

```
</div>
<div data-skcomponent="skerror" class="error-message" data-skvisibility="" ></div>
<div data-skcomponent="skerrormessage" class="error-message" data-skerrorid="email"></div>
<div data-skcomponent="skerrormessage" class="error-message" data-skerrorid="password"></div>
</div>
```

6.3 Apply and close the dialog to save your changes to the node.

7 Deploy this node in preparation for testing.

8 In the browser tab with your application, click **Register** button:

- 8.1 Enter name and age and at the email and password form enter invalid entries, or you can just leave them empty and click next:

- 8.2 Click the **x** to close the modal popup.

9 Continue with the next section.

2.7 Email notification to created user

Once the account has been created, it would be nice to send the user an email welcoming them to our application.

You can achieve this quickly by creating a notification template within PingOne and then triggering the template from your registration flow.

In this exercise you will also replace the send email to administrator from the earlier exercise with the send email to the new user instead. Unless you really like getting those administrator email messages.

- 1 In a browser tab **open** the **PingOne** console, if not already open.
 - 1.1 Click **User Experience** from the left menu.
 - 1.2 Click **Notification Templates** from the left menu.
 - 1.3 Click **+** icon at the top to create a new template.
 - 1.3.1 Select **General** in the **Type** field.
 - 1.3.2 Type **Welcome New User** for the **Name** field.
 - 1.3.3 Click **Create** button to create the template.
 - 1.4 In the **Email** section of the template:
 - 1.4.1 Click the **pencil** icon in the **Subject** field.
 - 1.4.2 Type **Welcome to Beginner's Luck**
 - 1.4.3 Click the **✓** icon to save the change to the **Subject** field.
 - 1.4.4 Click the **pencil** icon in the **New Email** field.
 - This is the body of the email you want to send to the user.

1.4.5 Copy and paste the following text into the field.

```
Hello ${given}!

<p>Congrats on creating your Beginner's Luck Online Gaming and Casino account!</p>

<p>May the odds be ever in your favor and please game responsibly!</p>

<p>To sign in, click <a href="${url}">here</a></p>
```

1.4.6 Click the ✓ icon to save the change to the **New Email** field.

- Note the `${}` blocks in the body, these are variables that you will define when you invoke the template in DaVinci.

1.5 The template is defined, since you will only be using this for email.

Email

Email templates support plain text or HTML.

From Name

From Address

Subject

Welcome to Beginner's Luck

New Email

Hello \${given}!

<p>Congrats on creating your Beginner's Luck Online Gaming and Casino account!</p>

<p>May the odds be ever in your favor and please game responsibly!</p>

<p>To sign in, click here</p>

1.6 Click the **x** to close the template dialog.

2 In a browser tab **open** the **DaVinci** console, if not already open.

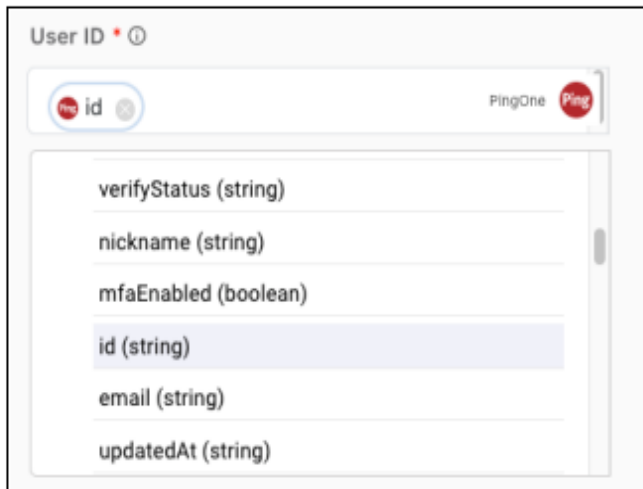
2.1 Click on the **Notify Admin** node at the end of your flow.

- This is what sends the administrator email message, you are going to change it to send an email instead in parallel to displaying the final page to the user.
- The same capability in the connector can be used just need to adjust the information for the user that was just created.

2.2 **Delete** the existing value in the **User ID** field.

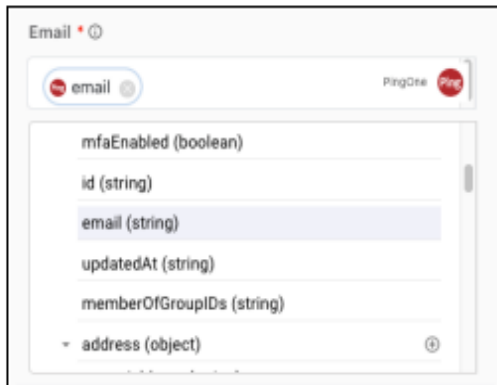
2.3 In the **User ID** field select the variable **id** from the **Create User** node.

- Ensure that you select the correct id (under user) in the list of variables.
- You can collapse the attributes under user that have sub-attributes by clicking the icon, to the left of it.



2.4 **Delete** the existing value in the **Email** field.

2.5 In the **Email** field select the variable **email** from the **Create User** node.



2.6 In the **Notification Name** field select **Welcome New User**

- This is the template you created and want to use.

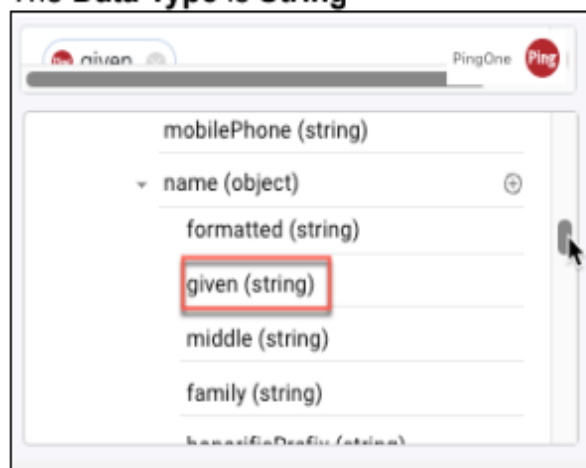
2.7 Now you need to define the variables that are used in the template, **scroll** down to **Notification Variables**.

2.7.1 Click **Add** button to add a variable:

2.7.1.1 **Variable Name** as given

2.7.1.2 In the **Value** field select the variable **given** from the **Create User** node, this will be under **name**.

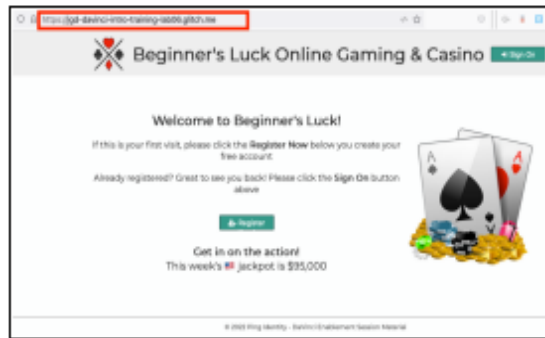
2.7.1.3 The **Data Type** is **String**



2.7.2 Click **Add** button to add a variable:

2.7.2.1 **Variable Name** as **url**

2.7.2.2 In the **Value** field you want to copy the URL for your web application from the address bar on that browser tab.



■ Of course this will be the URL for your application not the author's.

2.7.2.3 The **Data Type** is **String**

Input Variable 2:

Variable Name

url

Value

<https://gd-davinci-intro-training-lah06.nlitch.me/>

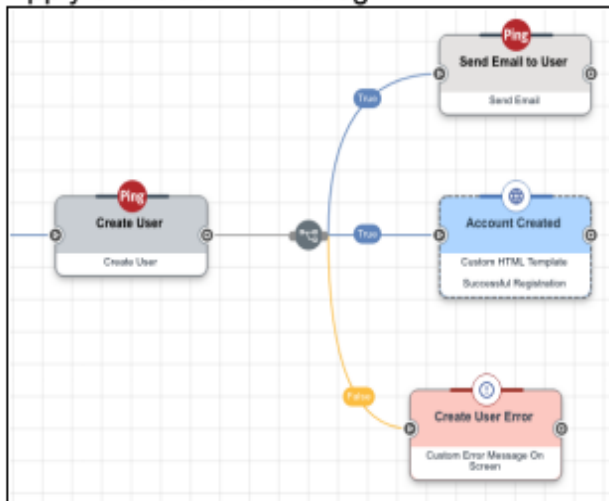
Data Type

String

2.8 Click **Apply** to save your changes to this point.

2.9 Select **Settings** and set **Node Title** to **Send Email to User**

2.10 Apply and close the dialog.

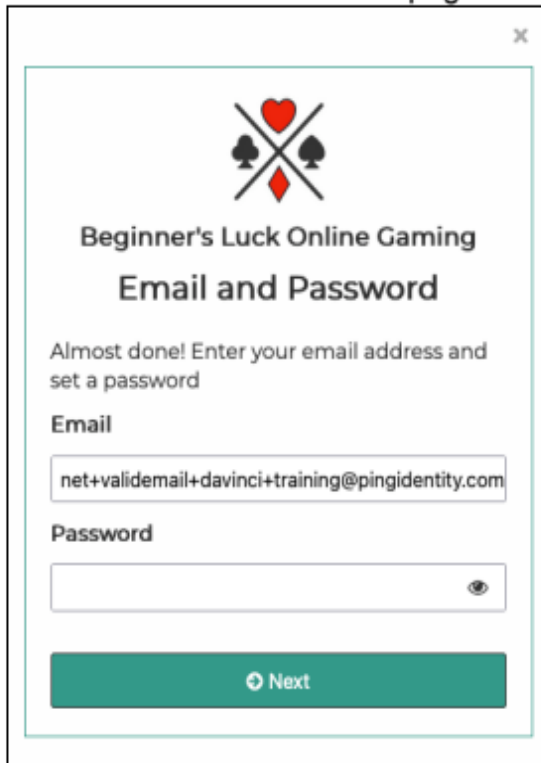
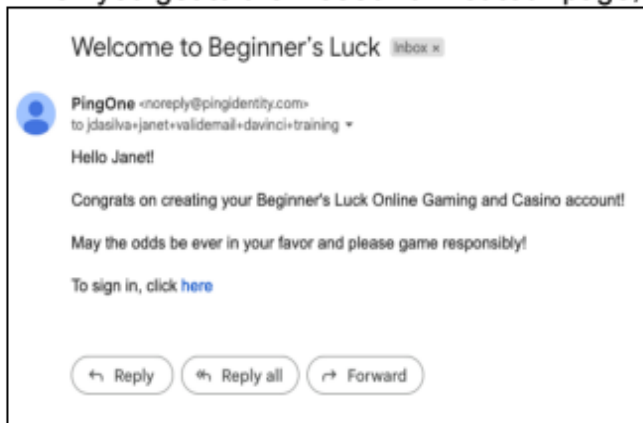


3 **Deploy** your flow.

4 Now go to the browser **tab** for your web application.

- In order to fully test this change you will need to use an email address that you can receive emails at.
- You can use a service such as <https://www.mailinator.com/> for this or
- Depending on your email provider be able to define alias emails

4.1 Click the **Register** button to create a new account.

4.2 At the Email and Password page enter an email that you can receive emails**4.3 Enter a valid password, remember it must contain at least at least one number, one letter lower case, one letter upper case, and a special character and be between 8 and 24 characters in length.****4.4 When you get to the Account Created! page, check your email inbox.**

5 The link in the email of course should take you to your application.

6 You have completed this exercise.