#CLESFSat 1/21/23

Welcome to Cleveland Salesforce Saturday! Thank you for joining us. Below is a recap of what happened.

RPG Project Buildout
You are encouraged to try building this project out in your own Trailhead Playground or Developer Org. We start working on the Character creation flow as outlined below and on the mural board (https://app.mural.co/t/clesfsat8113/m/clesfsat8113/1670075420334/e59b820e7ce6ebd059080222a700b5b8d6886d9b?sender=u957f7be550b3e9f0bf759019)

Today we add to the flow where we roll the die and set the 5 character attributes.

Flow
Open the Create Character Screen Flow we began last week.

1. Add a 3rd Screen element. Give it the label “Ready To Roll” and API Name Ready_to_Roll.
    a. Add a Display Text and give it the API Name messageRollPrep.
    b. Add a message congratulating the user on their {!Type} choice (include the resource), tell the user they are about to roll a 20-sided die, and then they will pick which attribute to apply the die roll to.
    c. Format this text.
2. Save the Flow (while not listed, you should do this after every numbered step).
3. Add an Action element. In the Action box start typing “Random” and select the Random20SidedDieController when you see it (if you don’t see it, you need to complete the steps from our 1/14 session document).
    a. Set the Label to “Roll the Die” and the API Name will be Roll_the_Die.
    b. Expand the Advanced section at the bottom of the screen and check the box for Manually assign variables.
    c. Under #output, create a New Resource.
        i. Resource Type: Variable
        ii. API Name: dieRoll
        iii. Data Type: Number
        iv. Decimal Places: 0
        v. Click Done
    d. Click Done
4. Add a 4th Screen element. Give it the Label “Roll Results” and API Name Roll_Results.

a. Add a Display Text and give it the API Name messageRollResult
b. Add a message telling them what they rolled (include the dieRoll variable) and ask them to pick one of the attributes below before clicking Next.
c. Format this text.
d. Add a Checkbox.
    i. Give it the label “Experience”.
    ii. Change the API Name to “choseExperience”.
    iii. Expand the Provide Help section and add a message telling the user what this attribute represents.
e. Repeat step d until you have checkboxes for all 5 attributes.

Label	**API Name**	**Help Message**
Experience	choseExperience	Street Smarts or life experience
Charisma	choseCharisma	Ability to influence or compel others
Well-Being	choseWellBeing	Health and Wellness
Smarts	choseSmarts	Book Smarts or IQ
Dexterity	choseDexterity	Flexibility and agility

5. Add a Decision element. Give it the Label “Which Attribute?” and the API Name will be Which_Attribute.
    a. There will be 5 outcomes, add them each with a single criteria (we will revisit and update this in a later step).

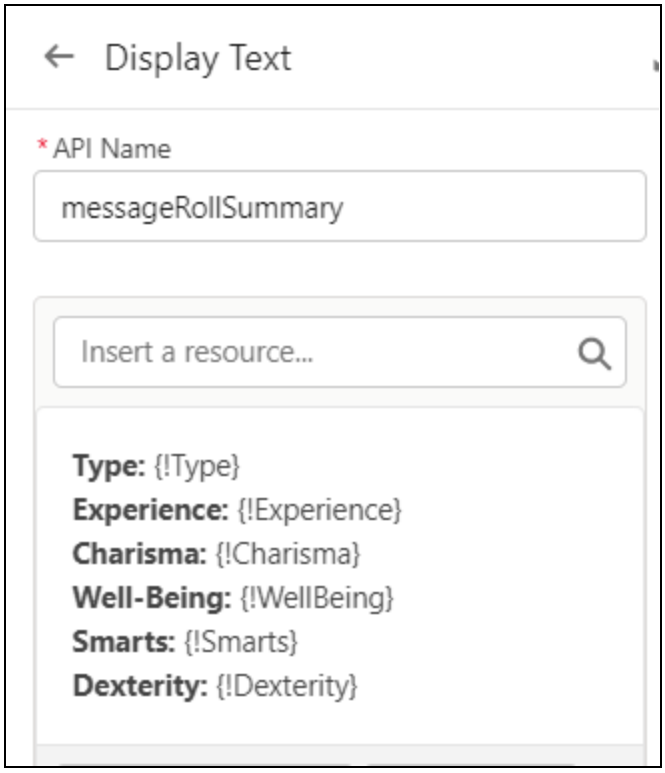
Outcome (Label/API Name)	Criteria
Experience Picked / Experience_Picked	{!choseExperience} Equals {!$GlobalConstant.True}
Charisma Picked / Charisma_Picked	{!choseCharisma} Equals {!$GlobalConstant.True}
WellBeing Picked / WellBeing_Picked	{!choseWellBeing} Equals {!$GlobalConstant.True}
Smarts Picked / Smarts_Picked	{!choseSmarts} Equals {!$GlobalConstant.True}
Dexterity Picked / Dexterity_Picked	{!choseDexterity} Equals {!$GlobalConstant.True}

6. For each Decision element (except the Default Outcome) you will add an Assignment element. The table below provides the details. In each one you will create the listed Resource as a New Resource with the Resource Type of Variable, the Data Type of Number and Decimal Places of 0.

Decision Route	Assignment Label	Resource	Operator & Value
Experience Picked	Assign Experience	Experience	Equals {!dieRoll}
Charisma Picked	Assign Charisma	Charisma	Equals {!dieRoll}
WellBeing Picked	Assign WellBeing	WellBeing	Equals {!dieRoll}
Smarts Picked	Assign Smarts	Smarts	Equals {!dieRoll}
Dexterity Picked	Assign Dexterity	Dexterity	Equals {!dieRoll}

7. Just above the End, click the plus and add another Decision element. Give it the Label “All Attributes Set?” and the API Name should be “All_Attributes_Set”.
    a. There will be 1 outcome with 5 criteria that check the variables created and assigned in Step 6. Give this outcome the label “All Attributes” and the API Name All_Attributes.
    b. The Condition is AND.
    c. The 5 criteria are:
        i. {!Experience} Is Null {!$GlobalConstant.False}
        ii. {!Charisma} Is Null {!$GlobalConstant.False}
        iii. {!WellBeing} Is Null {!$GlobalConstant.False}
        iv. {!Smarts} Is Null {!$GlobalConstant.False}
        v. {!Dexterity} Is Null {!$GlobalConstant.False}
8. Save the Flow (especially if you haven’t in a while).
9. From the All Attributes Outcome, add a Screen Element. Give it the Label “Roll Recap” and API Name Roll_Recap.
    a. Add a Display Text with the API Name of messageRollDone.
    b. Add a message congratulating the user on their die rolling and making their attribute decisions. Mention the results are presented and to click Next when finished reviewing them.
    c. Add a 2nd Display Text with the API Name of messageRollSummary.

d. Add a message that shows each variable result. Be sure to label them in a way the user will understand and include the correct resources. The image below shows one example.



10. Save the flow.
11. Return to the Roll Results Screen element created in Step 4. We need to update each Checkbox to only display if the similarly named variable is null. This way when we return to this screen multiple times the user will only be able to select an attribute that does not already have a value assigned.
    a. Click on the Experience checkbox.
    b. Expand Set Component Visibility.
    c. Change When to Display Component to “All Conditions are Met”
    d. In the box that appears, select the variable {!Experience} as the Resource, Operator as Is Null and Value as {!$GlobalConstant.True}.
    e. Repeat this for the other 4 checkboxes choosing the Resource as follows (but always setting the Operator to Is Null and the Value to {!$GlobalConstant.True}.
        i. Charisma: use {!Charisma}
        ii. Well-Being: use {!WellBeing}
        iii. Smarts: use {!Smarts}

iv. Dexterity: use {!Dexterity}

12. Save the flow.
13. Return to the Which Attribute? Decision element. We need to update this since we cannot reset the screen choices back to false. Instead, we will update each criterion to check that the current attribute level is null.
    a. Click on the Experience Picked outcome.
    b. Click the Add Condition button.
    c. Set the 2nd Condition to {!Experience} (this is the variable we created for the assignment) Is Null {!$GlobalConstant.True}.
    d. Repeat this for each Outcome.
        i. Charisma Picked: add {!Charisma} Is Null {!$GlobalConstant.True}
        ii. WellBeing Picked: add {!WellBeing} Is Null {!$GlobalConstant.True}
        iii. Smarts Picked: add {!Smarts} Is Null {!$GlobalConstant.True}
        iv. Dexterity Picked: add {!Dexterity} Is Null {!$GlobalConstant.True}
14. Save the flow.
15. Switch from Auto-Layout to Free Form so we can fix where Default Outcomes take us in our flow. (Don't worry, we'll turn it back on when we're done).
16. From the Which Attribute? Decision element, locate the Default Outcome line. Click on it and then delete it.
17. Then drag from the circle at the bottom of the Which Attribute? Decision element to the Roll Result Screen element (make sure the outcome is the Default Outcome).
18. From the All Attributes Set? Decision element, drag from the circle at the bottom to the Roll the Die Apex Action element (make sure the outcome is the Default Outcome).
19. Save the flow.
20. Switch back to Auto-Layout. Confirm that the Default Outcome from the Which Attribute? Decision shows that it goes to Roll Results and that the Default Outcome from the All Attributes Set? Decision shows that it goes to Roll the Die.
21. Save the flow.
22. Click the Debug button to test the flow.

In 2 Weeks: Naming our character and adding an image to finish out creating the character.