plotly

Star 23,446

Dash Python > ***Determining Which Callback Input Changed***

Determining Which Callback Input Changed

Sometimes you'll have multiple callback inputs and want to update an output differently depending on which input triggered the callback.

You can use the properties of `dash.callback_context` (or `dash.ctx` for short in Dash 2.4 and later) to determine which input triggered a callback.

> The following examples use `dash.callback_context.triggered_id` and `dash.callback_context.triggered_prop_ids` introduced in Dash 2.4.

Capturing the Component ID that Triggered the Callback

In this example, we get the `id` of the button that triggered the callback and display it each time a button is clicked. On first load, when no user action has triggered the callback, the value of `triggered_id` is `None`. We check for this and use it to display the message: "You haven't clicked any button yet".

> Note: `triggered_id` is available in Dash 2.4 and later. In earlier versions of Dash, you can get the triggered id with `dash.callback_context.triggered[0]['prop_id'].split('.')[0]`

```
from dash import Dash, html, dcc, Input, Output, ctx, callback

app = Dash()

app.layout = html.Div([
    html.Button('Button 1', id='btn-1-ctx-example'),
    html.Button('Button 2', id='btn-2-ctx-example'),
    html.Button('Button 3', id='btn-3-ctx-example'),
    html.Div(id='container-ctx-example')
])


@callback(Output('container-ctx-example','children'),
              Input('btn-1-ctx-example', 'n_clicks'),
              Input('btn-2-ctx-example', 'n_clicks'),
              Input('btn-3-ctx-example', 'n_clicks'))
def display(btn1, btn2, btn3):
    button_clicked = ctx.triggered_id
    return html.Div([
        dcc.Markdown(
            f'''You last clicked button with ID {button_clicked}
            ''' if button_clicked else '''You haven't clicked any button yet''')
    ])

if __name__ == '__main__':
    app.run(debug=True)
```

BUTTON 1 BUTTON 2 BUTTON 3

You haven't clicked any button yet

With Pattern-Matching Callback IDs

In the above example `triggered_id` is a string as the `id` of the input is a string. In cases where a component's `id` is a dictionary, for example, with **Pattern-Matching Callbacks**, `triggered_id` will also be a dictionary.

For example, with this component:

```
dcc.Dropdown(
        ['NYC', 'MTL', 'LA', 'TOKYO'],
        id={
            'type': 'filter-dropdown',
            'index': 0
        }
```

That is used as an input to a callback:

```
Input({'type': 'filter-dropdown', 'index': ALL}, 'value')
```

`triggered_id` returns:

```
{'index': 0, 'type': 'filter-dropdown'}
```

Here you can capture the index with: `triggered_id.index`, or the `type` with `triggered_id.type`.

Determining Multiple Input Triggers and Properties

In cases where multiple inputs trigger a callback, you can use `triggered_prop_ids`. It is also useful if different properties of the same component can trigger the callback and you want to capture the property.

`triggered_prop_ids` returns a dictionary of inputs that triggered the callback. Each key is a `<component_id>.<component_property>` and the corresponding value is the `<component_id>`.

In our first example above, if we used `triggered_prop_ids`, when Button 2 is clicked, it would return:

```
{'btn-2-ctx-example.n_clicks': 'btn-2-ctx-example'}
```

With Pattern-Matching Callbacks

For the pattern-matching callbacks 'filter-dropdown' example above, `triggered_prop_ids` returns:

```
{'{"index":0,"type":"filter-dropdown"}.value': {'index': 0, 'type': 'filter-dropdown'}}
```

With Pattern-Matching Callbacks With Multiple Triggers

If we have a series of dropdowns with different indexes (0-2) and they all triggered the callback, `triggered_prop_ids` returns:

```
{'{"index":0,"type":"filter-dropdown"}.value': {'index': 0, 'type': 'filter-dropdown'}, '{"inde
```

> In all of these examples, the dictionary keys are in the format `<component_id>.<component_property>, while the dictionary values are the component ids.

Dash Python ***> Determining Which Callback Input Changed***