A small guide to adding features

The interactive features are implemented using callbacks, which are a key feature of the dash library. Dash-callbacks consist of 3 parts: Inputs, Outputs and a Function. A simple example is shown in Figure 1.

```
@app.callback(
    # displays info when hovering over a line, info stays displayed until user hovers over another line
    Output(component_id='hover-info-line',component_property='children'),
    Input(component_id='net map',component_property= 'mouseoverEdgeData')
    )
def displayHoverEdgeData(data):
    if data:
        line_index_str = data['label']
        line_index = int(line_index_str)
        loading = round(loading_values[line_index],1) # uses global variable loading values to acces the loading percentage of a line
        return "line " + line_index_str  + " - Loading: :" + str(loading) + "%"

```

Inputs and Outputs need 2 parameters: component_id and component_property. **Component_id** is used to refer the component you want to use in the input/output. **Component_property** refers to which property of this component will be used as the input/output. (components and their properties are defined in the app.layout function). The property defined as input will be used as the input of the function. The property defined as output will have its value changed to the returned output of the function.

When more than one input or output is used, the inputs and the outputs of the function will have the same order as in which they were defined.

A short explanation of the example function.

Input: ‘net map’ refers to the network plot. ‘Mouseoveredgedata’ refers to the data contained in the edge the mouse is currently hovering over. This data is defined when generating the edge. The function uses this data as its input.

Function: first the function checks if the mouse is currently hovering over a line, by using “if data: ” . Then it generates a string which contains the index of the line and the loading percentage of the line. This string is then returned as the output of the function.

Output: ‘hover-line-info’ refers to a html.P component, which displays a string on the webpage. The component_property ‘children’ contains the string that will be displayed. This property will be changed to the returned output of the function.

A single component_property can be used as a input in different callbacks, but a single component_property can only be used as an output once. This can lead to very extensive callback-functions when a lot of features are required.

An example of this is the callback which has the component_property “elements” (of component ‘net map’, which refers to the network map). This callback handles all the changes made to the map, except style wise, which are handled in a different callback.

When you want to add a feature which makes changes to the map, this will have to be done within this callback. (Only changing the size or displaying a label can be realized in the callback which changes the stylesheet.)

Adding a feature that doesn’t change the map can be done by creating an entirely new callback.