plotly

Star 23,447

Dash App Life Cycle

This section describes the lifecycle of a Dash app.

1. When `python app.py` or `gunicorn app:server` is run, all of the files in a Dash app are executed. This means that if you have a statement such as `df = pd.read_csv('...')`, it is run when the program starts, rather than when the page is loaded. Therefore, if the CSV changes after the program starts, `df` will not be updated until the program restarts. In this case, it is recommended to provide data via a periodic task or by setting `app.layout` as a function to regenerate the layout with each page load.

2. When the page loads and the `dash-renderer` starts, it asks the server for both the initial layout (`app.layout`) and a list of all the callbacks registered to the app. This list of callbacks is only provided at page load and can't be modified later, so the app needs to know about all of its callbacks from the beginning even for components that do not exist in the initial layout.

    - Note that this is a recursive process: the `dash-renderer` collects not only those callbacks whose inputs can be changed directly through user interaction, but also those callbacks whose inputs are outputs of another callback that has already been collected. It's important that the `dash-renderer` collects the entire callback chain up front, or else it wouldn't be able to determine which callbacks are blocking others.

3. All callbacks that have inputs currently in the app layout are executed. This is known as the "initial call" of the callbacks, and this behavior can be suppressed by using the `prevent_initial_call` attribute.

    - If a Dash app has multiple callbacks, the `dash-renderer` requests callbacks to be executed based on whether or not their inputs might be changed as a result of another callback.

4. Components in the layout communicate with the `dash-renderer` whenever their state changes. When this occurs, the `dash-renderer` looks to see which callbacks need to be executed as a response to the user input. This can include both callbacks that use the input directly and callbacks whose inputs are outputs of callbacks that use the input directly.

- Since the `dash-renderer` has introspected the entire callback chain, it can delay the execution of callbacks whose inputs are outputs of callbacks that use the input directly until after callbacks that use the input directly have executed. This minimizes the number of requests the `dash-renderer` needs to make to the server in response to a particular user input.

5. It is possible for new components to be added to the `app.layout` dynamically as the output of a callback. If these new components are themselves inputs to callback functions, then their appearance in the layout triggers the execution of those callbacks.