Reactml

1. Define custom reusable components om the fly
2. Define a new component as composition of existing comps
3. Define a new component as a list of existing comps
4. Combinations of 2 and 3
5. Use existing (curated) comps from a dictionary
6. Add new components to such dictionaries
7. Setup with single command (assuming docker or npm)

Language

Each specification file consists of one or more UI component definitions. The definition can be a liat of components or a composition of other components. Each definition contained the name of the component being defined and a definition expression.

Definition s are separated by semicolons.

To define a component as a list of other components list them in order, separated by commas. For example:

page = header, body, footer:

body = breadcrumbs, dashSummary, dashRow1,dashRow2;

footer = copyright;

…

The above spec defines new react components called page, body and footer, as lists of other components, which maybe defend within the spec or defined externally by imported libraries or as code within the application.

Hierarchy

A component maybe defined as a hierarchy of other components, by separating them by arrows. For example:

Link = div -. span -.  a -. span -. “Click me”

Data specifiers

Dictionaries

New Component Name

Example:

**Page** = div(style={"color":"blue"}) -> Greeting;

Grouping with parentheses

Greeting =

**("Greetings ", (u->"from"), " "),**

BoldName,

(div**(style={"width":"300px", "height":"300px"})** -> Pie),

hr,

(div(style={"width":"300px", "height":"300px"}) -> Candlestick);

BoldName = strong -> "RTML!!";

Tag attributes – each value is a JSON value

Pie =

(h5 -> "this is a pie chart:"),

(div ->

VictoryPie(

cornerRadius=9

padAngle=3

data=[

{ "x": "Cats", "y": 35 },

{ "x": "Dogs", "y": 40 },

{ "x": "Birds", "y": 55 }

]

animate={"duration": 2000}

colorScale=["tomato", "orange", "gold", "cyan", "navy" ]

)

)**;**

Each definition ends in semicolon

Voronoi =

div(domain={ "x": [0, 5], "y": [0, 7] }) ->

VictoryVoronoi(

style={ "data": { "stroke": "#642aa1", "strokeWidth": 3 } }

data=[

{ "x": 1, "y": 2 },

{ "x": 2, "y": 3 },

{ "x": 3, "y": 5 },

{ "x": 4, "y": 4 },

{ "x": 5, "y": 7 }

Arrow defines hierarchy, comma for list

]);

Candlestick =

(h5 **->** "this is a candlestick:")**,**

VictoryCandlestick(data=[

{ "x": 1, "open": 5, "close": 10, "high": 15, "low": 0 },

{ "x": 2, "open": 5, "close": 10, "high": 20, "low": 0 },

{ "x": 3, "open": 5, "close": 10, "high": 22, "low": 0 },

{ "x": 4, "open": 5, "close": 10, "high": 18, "low": 0 },

{ "x": 5, "open": 5, "close": 20, "high": 25, "low": 0 },

{ "x": 6, "open": 5, "close": 10, "high": 12, "low": 0 },

{ "x": 7, "open": 5, "close": 10, "high": 19, "low": 0 }

]);