

Dash Introduction

Robert Heise und Florian Edenhofner – Education4Industry GmbH

17.03.2022

University4Industry

Agenda

Agenda

0. Was ist Plotly Dash?
1. Layout
2. Callbacks
3. Zusatz: Bootstrap

0. Was ist Plotly Dash?

0. Was ist Plotly Dash?

Dash basiert auf Flask, Plotly.js und React.js...

...man muss jedoch kein HTML, CSS oder Javascript können, um eine Dash App zu erstellen.

Es bietet:

- point-&-click interface, was klassische “dashboards” nicht bieten können
- low-code development direkt in Python
- einfaches deployment und scaling

1. Layout

1. Layout

Das Layout legt fest, wie die Dash App aussehen soll. Es besteht aus verschiedenen Komponenten:

1.1. Dash html components

1.2. Dash core components

1.3. Plotpy Graphs

1.1. Dash html components

```
1  from dash import html
```

HTML Komponenten bilden die Basis für die Elemente in einer Dash App. Dadurch können die klassischen HTML Komponenten, wie Text oder Buttons, abgebildet werden.

Beispiel für html component

```
1  app = dash.Dash()
2
3  app.layout = html.Div(
4      children=[
5          html.H1(id = 'H1',
6                  children = 'HTML Component',
7                  style = {'textAlign':'center','marginTop':40,'marginBottom':40}),
8      ]
9  )
10
11 if __name__ == '__main__':
12     app.run_server()
```


1.1. Dash html components - Weitere html components

```
1  html.Div()  
2  html.Button()  
3  html.H3(), html.H2(), html.H1()  
4  html.Img()
```

1.2. Dash core components

```
1  from dash import dcc
```

Core Komponenten erweitern die html components durch spezielle für Dash entwickelte Objekte.

Beispiel für eine core component

```
1  app.layout = html.Div(  
2      children=[  
3          dcc.Graph(id = 'line_plot', figure = stock_prices()),  
4      ]  
5  )
```

1.2 Weitere core components

```
1  dcc.Graph()  
2  dcc.Input()  
3  dcc.Dropdown()  
4  dcc.Slider()  
5  dcc.Textarea()  
6  dcc.Checklist()  
7  dcc.RadioItems()  
8  dcc.DatePickerSingle()
```

1.3. Plotly Graphs

```
1 import plotly.express as px
```

Plotly Graphs sind eine Bibliothek, die es vereinfacht verschiedene Plots aus Daten zu erstellen. Zum Beispiel: Map, Scatter, Line plots...

Beispiel für Plotly Graph

```
1 app.layout = html.Div(  
2     children=[  
3         dcc.Graph(id = 'line_plot', figure = px.line(df, x = df['x-label'], y = df['y-label']))  
4     ]  
5 )
```

2. Callbacks

2. Callbacks

```
1 from dash.dependencies import Input, Output
```

Wenn man dash components mit weiteren dash components oder Plotly Plots für interaktive Inputs verknüpfen möchte, werden Callbacks genutzt. Callbacks sind Python Funktionen, die aufgerufen werden, wenn sich ein 'input' ändert.

Connects Dash components + Plotly = Interactive Features

Beispiel für Callback

```
1 @app.callback(Output(component_id='line_plot', component_property= 'figure'),
2               [Input(component_id='dropdown', component_property= 'value')])
3 def graph_update(value_of_input_component):
4     print(value_of_input_component)
5     fig = px.line(df, x = df['date'], y = df[value_of_input_component])
6     return fig
```

3. Zusatz: Bootstrap

3. Zusatz: Bootstrap

```
1  import dash_bootstrap_components as dbc
```

Benutzen eines externen Themes:

```
1  app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])
```

Mit Dash Bootstrap können Dash Apps um Bootstrap CSS erweitert werden. Dadurch können fließende Objekte eingebaut werden, die angeben wo die Komponenten in der Dash App angezeigt werden.

Beispiel für Bootstrap

```
1  app.layout = html.Div(  
2      children=[  
3          dbc.Row([  
4              dbc.Col([html.H1('HTML Component')]),  
5              dbc.Col([html.H1('HTML Component')]),  
6              ], align='center'),  
7      ]  
8  )
```