

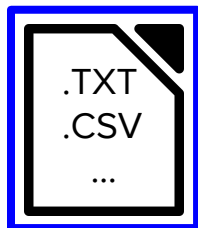


POLYTECH[°]
GRENOBLE

Dashboard generator for scientific applications

BADAT Leya
CUAU Victor
MASSON Jérémy

The process



+



+

Reg[ular] *
Ex[pression]

```
python Prototype7.py Data/TempHum.csv  
"^[a-zA-Z0-9\-.]*,(?1),(?1)" Templates/template.yaml
```



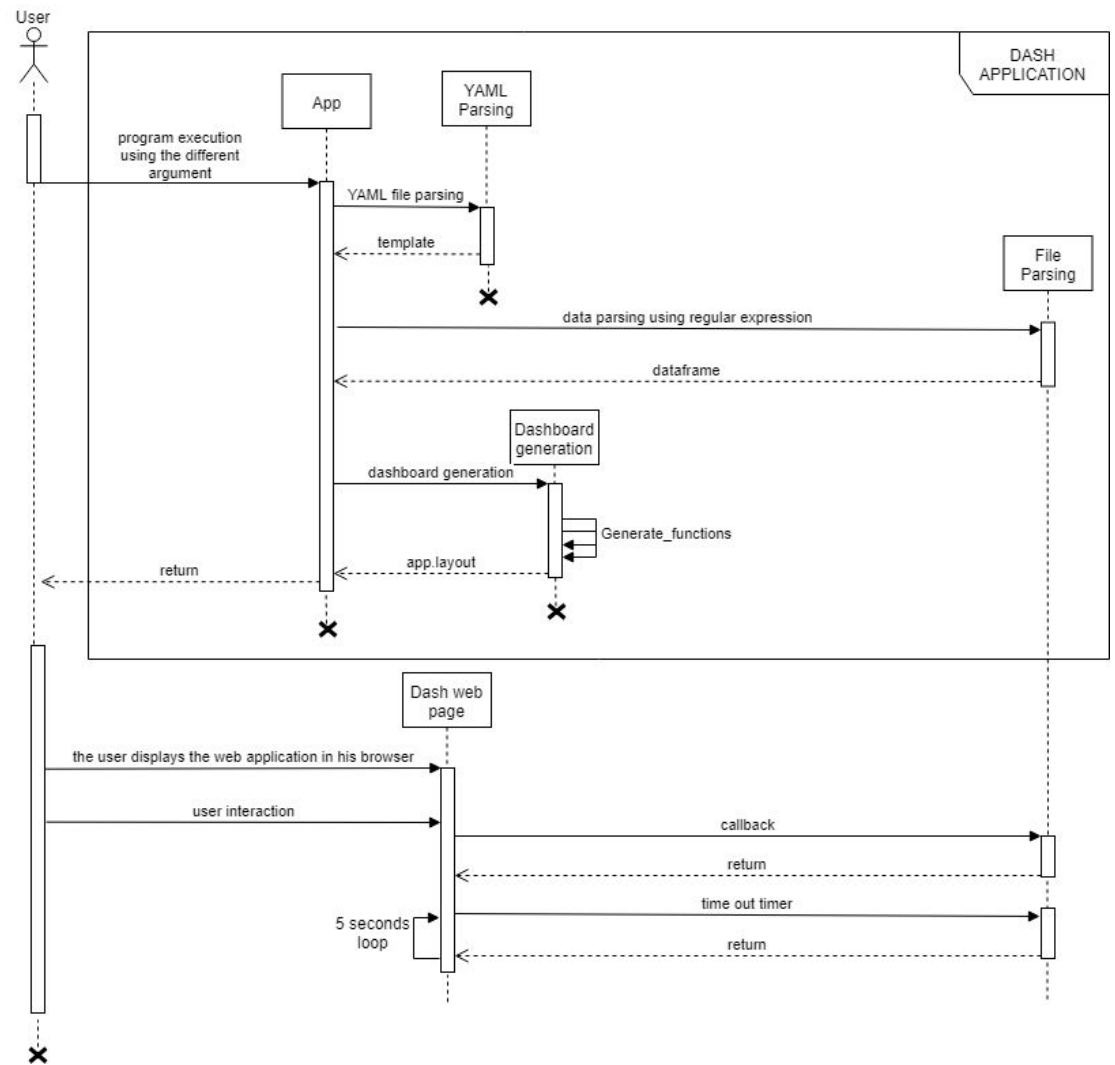
python™

Our program



<http://127.0.0.1:8050/>

Sequence diagram



Plotly Dash : Callbacks

```
@app.callback(  
    Output(component_id='my-div', component_property='children'),  
    [Input(component_id='my-id', component_property='value')]  
)  
def update(input_value):  
    return 'You\'ve entered {}'.format(input_value)
```

Output

name: my-div

field: children

Input

name: my-id

field: value

Update the output field
`update(input_value)`
return

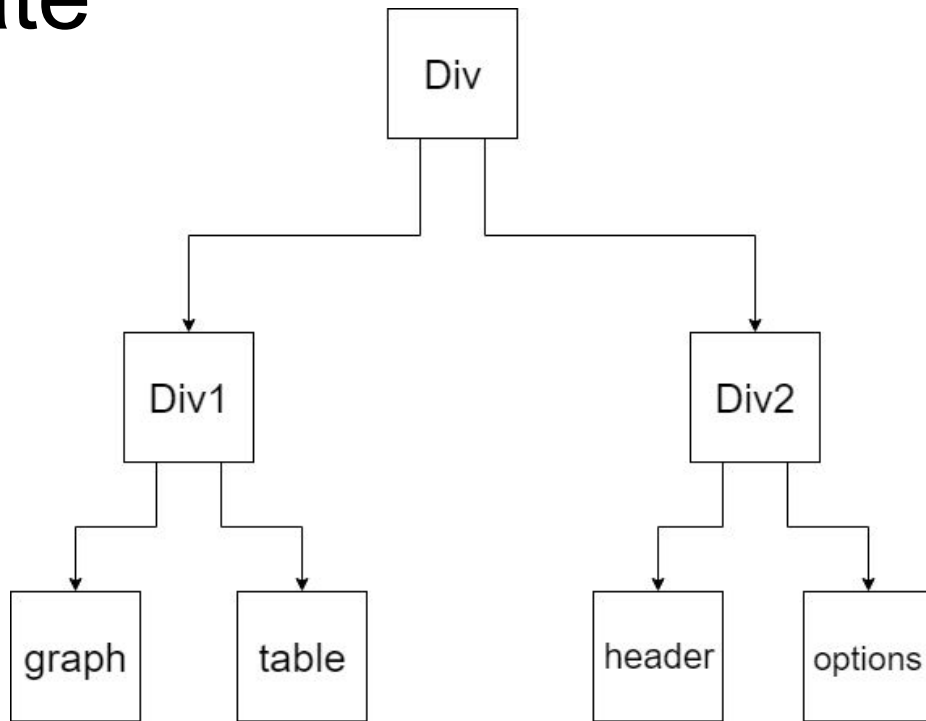
Input field modified

The YAML template

```
1 div:
2   # LOWER BOX
3   div1:
4     graph:
5     table:
6   # TOP BOX
7   div2:
8     header:
9     options:
```



python object
dictionary



Regular expressions

File :

```
#comment  
x y z  
1 5 9
```



Findall

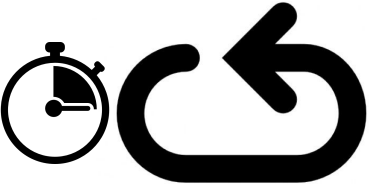
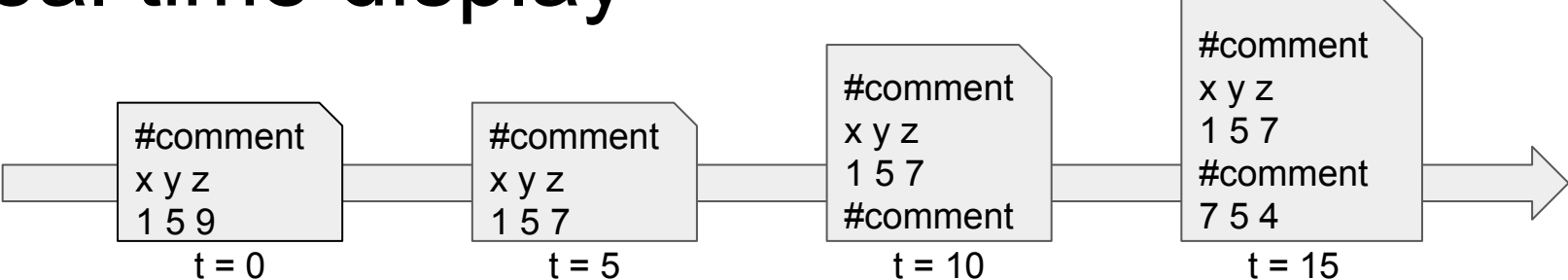


Search
(to be able to ignore
comments)



Version	Regular expression	Result
First version : basic regular expression	<code>^[a-zA-Z0-9]</code>	<code>x y z 1 5 9</code>
Second version : groups	<code>^([a-zA-Z0-9]*)\s[a-zA-Z0-9]*\s([a-zA-Z0-9]*)</code>	<code>x z 1 9</code>
Third version : recursivity	<code>^([a-zA-Z0-9]*)\s(?:)\s(?:)</code>	<code>x z 1 9</code>

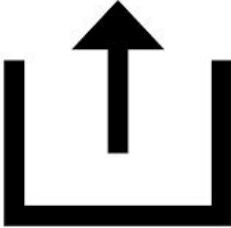
Real time display



Timer triggered every
5 seconds



The browser doesn't
have to be refreshed
manually



Values updated

Future improvements

- Embedding Dash apps inside a Flask app
- Creating a cache to store already uploaded YAML templates
- Enhancing real time display

➔ Python3

```
from dashboard_app import *
from cache import *

server = flask.Flask(__name__)

data_input = html.H1(id = 'user', children = '')

dash_app1 = Dash(__name__, server = server, url_base_pathname = '/dashboard/')
dash_app1.layout = html.Div(id = 'father')

@server.route('/')
@server.route('/help')
def render_help():
    # DISPLAY AN INTRODUCTION / HELP PAGE

@server.route('/dashboard', methods=['GET', 'POST'])
def render_dashboard():
    global server, dash_app1, data_input
    data = request.form['data']
    data_input = html.H1(id = 'user', children = data)
    # UPDATE THE DASHBOARD APP
    return dash_app1

@server.route('/realtimedashboard', methods=['GET', 'POST'])
def render_realtimedash():
    # UPDATE THE REAL TIME DASHBOARD APP
    return dash_app2

@server.route('/templates')
def render_cachedtemp():
    # DISPLAY THE CACHED TEMPLATES

app = DispatcherMiddleware(server, {
    '/dash1': dash_app1.server,
    '/dash2': dash_app2.server,
})

run_simple('0.0.0.0', 5000, app, use_reloader=True, use_debugger=True)
```




POLYTECH[°]
GRENOBLE

Thank you for your attention!

BADAT Leya ; CUAU Victor ; MASSON Jérémy

Polytech Grenoble 2018-2019, filière INFO4, sous la responsabilité de Olivier RICHARD