Python Dash App Tutorial

Dash is Python framework for building web applications. It built on top of Flask, Plotly.js, React and React Js. It enables you to build dashboards using pure Python. Dash is open source, and its apps run on the web browser. In this tutorial, we will show how to create a dash app and host it as a web application on Heroku cloud application platform

Prerequisites:

- **Python Anaconda**
- **Git**
- **Heroku cli**

There are many different ways to host a dash app on Heroku, this is just one of them.

Once these are installed you can use the anaconda prompt or terminal to install several dash libraries. These libraries are under active development, so install and upgrade frequently. Python 2 and 3 are supported.

```
pip install dash
pip install dash-renderer
pip install dash-html-components
pip install dash-core-components
pip install plotly --upgrade
```

Dash App Layout

Dash apps are composed of two parts. The first part is the "layout" of the app and it describes what the application looks like. The second part describes the interactivity of the application. Dash provides HTML classes that enable us to generate HTML content with Python. To use these classes, we need to import dash_core_components and dash_html_components. You can also create your own custom components using Javascript and React Js.

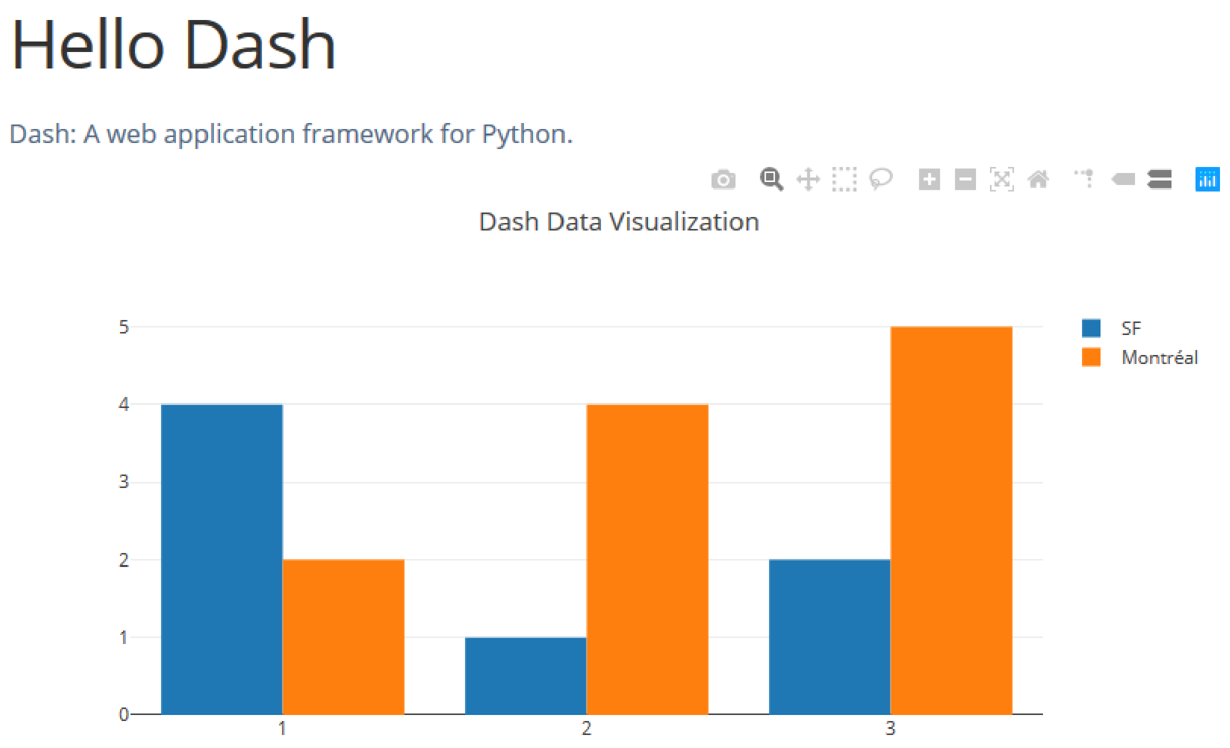
Lets create a file called app.py using any IDE(Py charm, Spyder etc.) or text editor

```
import dash
import dash_core_components as dcc
import dash_html_components as html

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__, external_stylesheets=external_stylesheets)

app.layout = html.Div(children=[
    html.H1(children='Hello Dash'),

    html.Div(children='''
        Dash: A web application framework for Python.
    '''),

    dcc.Graph(
        id='example-graph',
        figure={
            'data': [
                {'x': [1, 2, 3], 'y': [4, 1, 2], 'type': 'bar', 'name':
'SF'},
                {'x': [1, 2, 3], 'y': [2, 4, 5], 'type': 'bar', 'name':
u'Montréal'},
            ],
            'layout': {
                'title': 'Dash Data Visualization'
            }
        }
    )
])

if __name__ == '__main__':
    app.run_server(debug=True)
```

If you are using text editor, run the app in anaconda terminal with

python app.py

Or in case of IDE you can run it there and visit http://127.0.0.1:8050/ in your web browser. You should see an app that looks like this.



To know more about dash components you can visit these useful resources

- https://dash.plot.ly/getting-started
- https://dash-gallery.plotly.host/Portal/
- https://dash.plot.ly/

Here are some dash app repositories you can use

- https://github.com/vhegde13/dashapp-regression (Boston Housing Dataset Linear Regression) An app that provides basic EDA visualization and linear regression on the boston housing dataset. Click here to get the link to the app
- https://github.com/zwrankin/dash_tutorial (Dash Tutorial - Sustainable Development Goals)

You can create a copy of them in your github through forking or cloning or downloading

Hosting Dashboard on Heroku

Note: It is recommended that first you use a very simple app like the one mentioned below (hello dash) to deploy on Heroku, this can reduce your time to debug dependencies that could be introduced if you were deploying your own complex app.

Use conda prompt or any terminal, First, create a directory that will hold all your projects files. You can do this on Ubuntu using the mkdir command. For windows it will be the same

```
$ mkdir my_dash_app
```

```
$ cd my_dash_app
```

Next, initialize the folder with git(Install git) and virtualenv(pip install virtualenv). Git is for version control, and virtualenv will enable us to create a virtual environment to hold all our Python dependencies. After creating the environment, we activate it.

```
$ git init
```

```
$ virtualenv venv
```

```
$ cd venv
```

```
$ ./Scripts/activate
```

Next, we install all the packages we need for our Dash application:

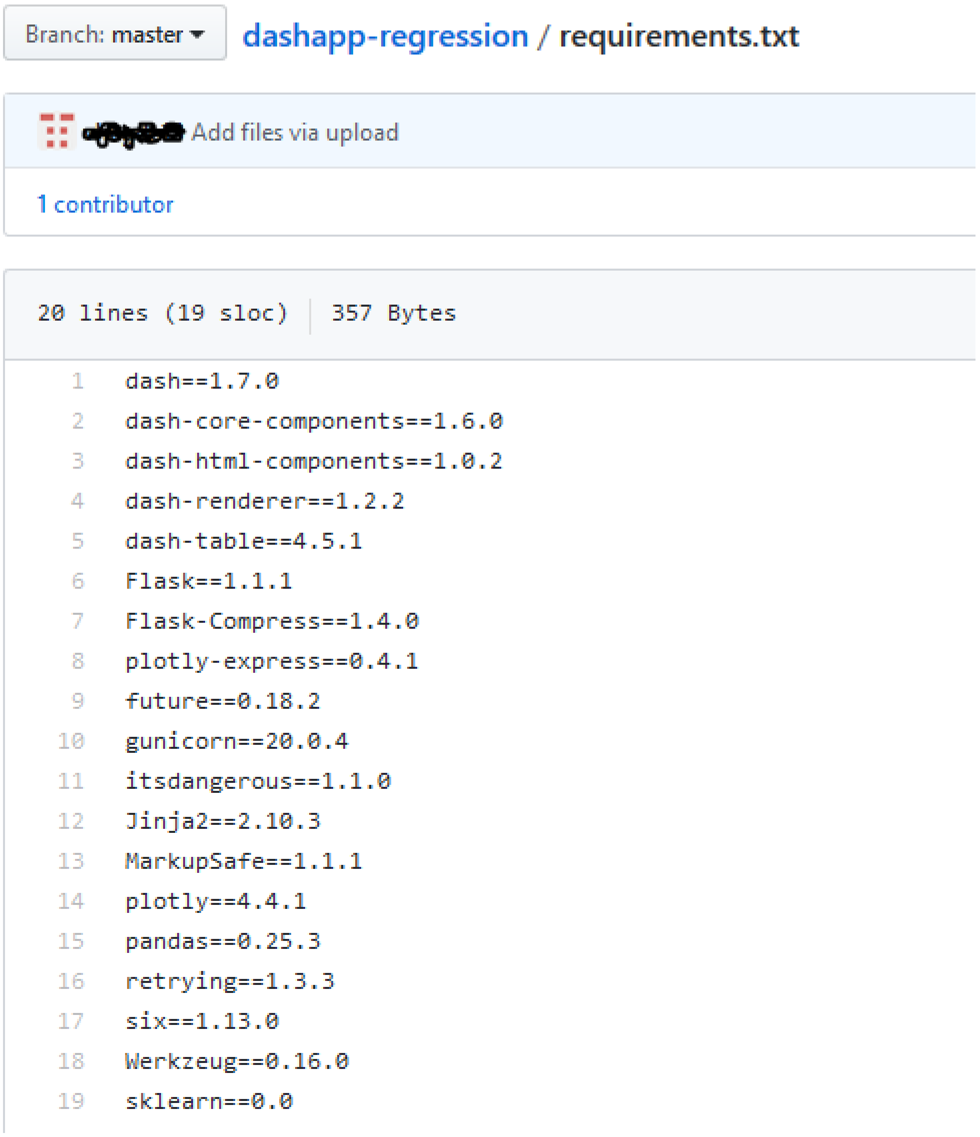
```
$ pip install dash
$ pip install dash-renderer
$ pip install dash-core-components
$ pip install dash-html-components
$ pip install plotly
```

In order to serve our Dash application, we need a Python web server. We never use Flask's development server in production. We use Gunicorn web server for this function. Install it as shown below:

```
$ pip install gunicorn
```

Next, we need to create a few files in our folder:

1. app.py where we will code our dash application.
2. A .gitignore to make sure that unnecessary files are not pushed to production (Created in Git or you can copy the file from the git repository links mentioned above. You can create your own, follow instructions at https://stackoverflow.com/questions/10744305/how-to-create-a-gitignore-file). A good gitignore file at https://github.com/github/gitignore/blob/master/Python.gitignore
3. A requirements.txt file that will contain all the Python dependencies and their versions. This file is for Heroku as it will install these libraries during deployment (Screenshot below of how it looks)


Branch: master ▾ dashapp-regression / requirements.txt

[redacted] Add files via upload

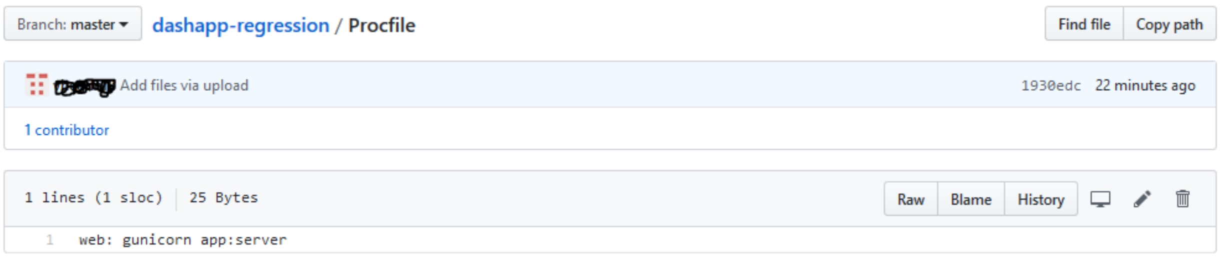
1 contributor

20 lines (19 sloc) | 357 Bytes

```
dash==1.7.0
dash-core-components==1.6.0
dash-html-components==1.0.2
dash-renderer==1.2.2
dash-table==4.5.1
Flask==1.1.1
Flask-Compress==1.4.0
plotly-express==0.4.1
future==0.18.2
gunicorn==20.0.4
itsdangerous==1.1.0
Jinja2==2.10.3
MarkupSafe==1.1.1
plotly==4.4.1
pandas==0.25.3
retrying==1.3.3
six==1.13.0
Werkzeug==0.16.0
sklearn==0.0
```

```
pip freeze > requirements.txt
```

4. A Procfile for deployment.


Branch: master ▾ dashapp-regression / Procfile

[redacted] Add files via upload 1930edc 22 minutes ago

1 contributor

1 lines (1 sloc) | 25 Bytes

```
web: gunicorn app:server
```

Add the following to app.py. This is just an example so you can use your own Dash dashboard.

```
import os
import dash
import dash_core_components as dcc
import dash_html_components as html
app = dash.Dash(__name__)
server = app.server
app.layout = html.Div([
    html.H2('Hello World'),
    dcc.Dropdown(
        id='dropdown',
        options=[{'label': i, 'value': i} for i in ['LA', 'NYC', 'MTL']],
        value='LA'
    ),
    html.Div(id='display-value')
])
@app.callback(dash.dependencies.Output('display-value', 'children'),
              [dash.dependencies.Input('dropdown', 'value')])
def display_value(value):
    return 'You have selected "{}"'.format(value)
if __name__ == '__main__':
    app.run_server(debug=True)
```

It is important to remind ourselves that at its core a Dash application is also a Flask application. For purposes of deployment, we need to access the Flask application instance. Dash enables us to do this using app.server

```
app = dash.Dash(__name__)
server = app.server
```

The next step is to create a Heroku app on the terminal and add all our application packages. Once you have committed the changes; push your application to heroku master (Require Heroku client and git mentioned in the prerequisite). The output of this command will have a link to your live Dash application on Heroku. Before this make sure you have Heroku cli and git installed and you will also have to create an account on Heroku and login so that you can push the app through your account.

```
$ heroku create your-app-name
$ git add .
$ git commit -m 'Your-commit-message'
$ git push heroku master
```

This should push the app to the Heroku cloud platform

Frequently Asked Questions

Error Trying to initialize Dash in Spyder IPython Console

https://stackoverflow.com/questions/49370550/error-trying-to-initialize-dash-in-spyder-ipython-console

Issue with virtualenv - cannot activate

https://stackoverflow.com/questions/8921188/issue-with-virtualenv-cannot-activate

Difference between forking and cloning a repository

https://github.community/t5/Support-Protips/The-difference-between-forking-and-cloning-a-repository/ba-p/1372

Heroku create/python/git/venv not recognized as an internal or external command (Windows)

https://stackoverflow.com/questions/42170691/heroku-not-recognized-as-an-internal-or-external-command-windows

Same will be the case for others too, check if the programs are installed and the environment variables have the right address

For other faq visit https://dash.plot.ly/faqs

References

https://towardsdatascience.com/exploring-sgs-rentals-with-an-interactive-webapp-built-using-dash-flask-and-heroku-part-1-cccd9e8dd1b8

zwrankin/dash_tutorial: Gentle introduction to dash development and deployment via Heroku

How to Build a Reporting Dashboard using Dash and Plotly

Bokeh vs Dash — Which is the Best Dashboard Framework for Python?

b&apos;Your First Dash App | alishobeiri | Plotly&apos;