モダンWebフロントエンド環境で作る Jupyter Widget

おのうえ (@_likr)

PyData Osaka Meetup #3

自己紹介

- おのうえ (@_likr)
- 京都大学 学際融合教育研究推進センター
  政策のための科学ユニット 特定助教
- ng-kyotoオーガナイザー、GDG神戸スタッフ
  関西フロントエンドUG、Kobe.py
- 可視化、最適化、アルゴリズムの研究

前回のあらすじ

NetworkXのグラフ描画はもうちょっとなんとかなるんじゃないかと思った

「NetworkXによるグラフの分析と可視化」@PyData Osaka Meetup #1

目的

- JupyterとWebベースのグラフ描画ライブラリを使って対話的なグラフ可視化を行う
- widget-cookiecutterを使用
- トランスパイラとモジュールバンドラを使ったモダンなフロントエンド開発スタイルを採用
- グラフ描画ライブラリはWebComponentsとして提供

Jupiter Widget

- http://jupyter.org/widgets.html
- https://ipywidgets.readthedocs.io/en/latest/
- Jupyter Notebookに対話的な要素を追加する
  - 対話的なデータ可視化
  - Ipyleaflet、bplot、pythreejs
- ZMQとWebSocketを使ったPython-JS間の値同期

WebComponents

- https://www.webcomponents.org/
- 独自の振る舞いをするHTML要素を作成し
  配布、再利用する仕組み
- 4つのWeb標準規格の総称
  - CustomElements、HTML Templates、
    Shadow DOM、HTML Imports
- WebComponentsを便利にするフレームワーク
  - Polymer、X-Tag、SkateJS、Bosonic

EgRenderer

- 自作グラフ描画ライブラリ
  - canvasベース
  - トランジションを頑張る
- めっちゃ作りかけ
  - WebGL対応したい
  - OGDFのレイアウトをサポートしたい
    - Emscripten (asm.js & WebAssembly)
- https://likr.github.io/eg-renderer-canvas/

HTML API

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>EgRenderer Example</title>
<script src="https://cdnjs.cloudflare.com/ajax/libs/webcomponentsjs/0.7.22/webcomponents.min.js"></script>
<link rel="import" href="../../eg-renderer.html">
</head>
<body>
<eg-renderer
  width="600"
  height="400"
  data='{
    "vertices": [
      {"u":0, "d": {"text": "source"}},
      {"u":1, "d": {"text": "target"}}
    ],
    "edges": [
      {"u": 0, "v": 1, "d": {}}
    ]
  }'
  transition-duration="500"
  layout="hierarchy">
</eg-renderer>
</body>
</html>
```

JS API

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>EgRenderer Example</title>
<script src="https://cdnjs.cloudflare.com/ajax/libs/fetch/1.0.0/fetch.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/webcomponentsjs/0.7.22/webcomponents.min.js"></script>
<link rel="import" href="../../eg-renderer.html">
</head>
<body>
<eg-renderer
  width="600"
  height="400"
  transition-duration="500"
  layout="hierarchy">
</eg-renderer>
<script>
fetch('data.json')
  .then((response) => response.text())
  .then((data) => {
    const renderer = document.querySelector('eg-renderer')
    renderer.setAttribute('data', data)
    renderer
      .layout()
      .center()
  })
</script>
</body>
</html>
```

widget-cookiecutter

- cookiecutter
  - https://github.com/audreyr/cookiecutter
  - プロジェクトテンプレート
  - Python製
- widget-cookiecutter
  - Jupyter Widget用テンプレート
  - PyPI、npmへの配布が容易

```
$ pip install cookiecutter
$ cookiecutter https://github.com/jupyter/widget-cookiecutter.git
```

Development

```
$ git clone https://github.com/likr/pyegrenderer.git
$ cd pyegrenderer
$ pip install -e .
$ jupyter nbextension install --py --symlink --sys-prefix pyegrenderer
$ jupyter nbextension enable --py --sys-prefix pyegrenderer
```

- Pythonコードはpyegrenderer(プロジェクト名)以下
- JSコードはjs以下
  - npmでモジュール管理
  - webpackで開発

トランスパイラとモジュールバンドラ

- トランスパイラ
  - JSの機能を拡張、古いブラウザへの対応
  - Babel、TypeScript
- モジュールバンドラ
  - 複数のJSファイル等を単一のファイルにまとめる
  - Node.jsスタイルのプログラミング(CommonJS)
  - Webpack
    - widget-cookiecutterに標準搭載
    - loaderでトランスパイルも可能

js/webpack.config.js

```
var loaders = [
  {
    test: /\.js$/,
    include: [
      path.resolve(__dirname, 'src'),
      path.resolve(__dirname, 'node_modules/eg-renderer-canvas')
    ],
    loader: 'babel-loader',
    query: {
      presets: ['latest'],
      plugins: [
        'transform-custom-element-classes'
      ]
    }
  },
  {
    test: /\.json$/,
    loader: 'json-loader'
  }
]
```

js/src/example.js

```
// Custom View. Renders the widget model.
const HelloView = widgets.DOMWidgetView.extend({
  render: function () {
    this.model.on('change:data', () => {
      const data = this.model.get('data')
      this.component.setAttribute('data', data)
    })

    this.model.on('change:layout_method', () => {
      const layoutMethod = this.model.get('layout_method')
      this.component.setAttribute('layout', layoutMethod)
    })

    this.component = document.createElement('eg-renderer')
    this.component.setAttribute('auto-update', '')
    this.component.setAttribute('auto-centering', '')
    this.component.setAttribute('transition-duration', '500')
    this.component.setAttribute('width', '1000')
    this.component.setAttribute('height', '750')
    this.component.setAttribute('data', this.model.get('data'))
    this.component.setAttribute('layout', this.model.get('layout_method'))
    this.el.appendChild(this.component)
  }
})
```

pyegrenderer/example.py

```python
import json
import ipywidgets as widgets
from traitlets import Unicode


@widgets.register('hello.Hello')
class EgRenderer(widgets.DOMWidget):
    """"""
    _view_name = Unicode('HelloView').tag(sync=True)
    _model_name = Unicode('HelloModel').tag(sync=True)
    _view_module = Unicode('pyegrenderer').tag(sync=True)
    _model_module = Unicode('pyegrenderer').tag(sync=True)
    data = Unicode('{"vertices": [], "edges": []}').tag(sync=True)
    layout_method = Unicode('hierarchy').tag(sync=True)

    def render(self, graph):
        vertices = [{'u': u, 'd': {'text': graph.node[u]['label']}}
                    for u in graph.nodes()]
        edges = [{'u': u, 'v': v, 'd': {}}
                 for u, v in graph.edges()]
        self.data = json.dumps({'vertices': vertices, 'edges': edges})
```

デモ

- https://youtu.be/0RvD9TlIrSo

まとめ

- グラフ可視化するJupyter Widget作った
  - https://github.com/likr/pyegrenderer
- widget-cookiecutterが便利だった
- モダンなフロントエンドフレームワークの利用も可能
- 今後も真面目に作るかは未定
- 既に本格的なグラフ可視化Widgetもあるよ!
  - https://github.com/cytoscape/jupyter-cytoscape