Jupyter for React.js developers

React.js components to build your custom data product with Jupyter

Lightning Talk at FOSDEM 2022
Saturday 5th February 2022

Eric Charles, founder of Datalayer, a data science platform, committer for JupyterLab and Jupyter Server
What is the issue?

- JupyterLab uses Lumino UI toolkit
  - Lumino is imperative
    https://github.com/jupyterlab/jupyterlab
    https://github.com/jupyterlab/lumino

- ISSUE: React.js can not wrap JupyterLab
  → Jupyter React Library solve this issue
JupyterLab is not React.js

The JupyterLab App is **NOT** React.js

It is a Lumino App and can not be used by React.js developers
Why is this an issue?

- You have to fallback to **iframe solutions** to show a composed UI **mixing your existing React UI with JupyterLab**
  - Those are **separated applications** which does not talk together

- You **need to learn yet-another UI toolkit (Lumino) to customize JupyterLab**
  - No “declarative” language
  - No reusable widgets
  - No out-of-the-box integration with de-facto state management systems like redux, mobx...
Jupyter React Library enables communication between Jupyter and your React.js App

Your React.js Application

Jupyter React

Send Commands to JupyterLab

Lumino Signals to React Redux Observables

JupyterLab

Classic Notebook
Jupyter React architecture

JupyterLab functionalities can now be used in any React.js application.

Lumino Command and Signals
<->
React Redux Observables
Usage

```javascript
<Jupyter terminals={true} collaborative={true}>
  <Notebook path="test.ipynb"/>
</Jupyter>
```

```javascript
export type INotebookProps = {
  path: string;
  ipywidgets?: "classic" | "modern";
  sidebarComponent: (props: any) => JSX.Element;
  sidebarMargin: number;
};
```
Jupyter React Components Example

© Datalayer, 2021

Lumino

IpyWidgets

Outputs

Cell

Notebook

Commands

Console

Dialog

File Browser

Settings

Terminal
Docusaurus Example

**Tutorial Intro**

Let's discover Docusaurus in less than 5 minutes.

```python
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0, 8, 50)
x2 = np.linspace(0, 2, 50)
y1 = np.cos(x1) + np.pi - x1 + np.exp(-x1)
y2 = np.cos(x2) + np.pi + x2
fig, (ax1, ax2) = plt.subplots(2, 1)
fig.suptitle('A tale of 2 subplots')
ax1.plot(x1, y1, 'g--')
ax1.set_ylabel('Damped oscillation')
ax2.plot(x2, y2, '-.')
ax2.set_ylabel('Undampend')
ax1.set_xlabel('time (s)')
plt.show()
```

Getting Started
Slate Example

Build a Google-Docs-like Notebook
Next Steps

- **Realtime Collaboration**
  - Available for notebooks (built-in jupyterlab collaboration)
  - For other components, use syncable stores (redux, mobx...)

- Integrate with **ProseMirror editor**
  - First with vanilla javascript plugins
  - Then with React.js components

- **Add reactivity** “a-la-observablehq”
  - [https://observablehq.com](https://observablehq.com)

- **Connect** with Jupyter community, developers and users interested in React.js
THANK YOU!

Any questions?

- [eric@datalayer.io](mailto:eric@datalayer.io)
- [@echarles](https://twitter.com/echarles)
- [https://twitter.com/echarles](https://twitter.com/echarles)
- [https://github.com/datalayer/jupyter-react](https://github.com/datalayer/jupyter-react)
- [https://github.com/datalayer/jupyter-examples](https://github.com/datalayer/jupyter-examples)
- [https://github.com/datalayer/jupyter-docusaurus](https://github.com/datalayer/jupyter-docusaurus)