

Jupyter for React.js developers

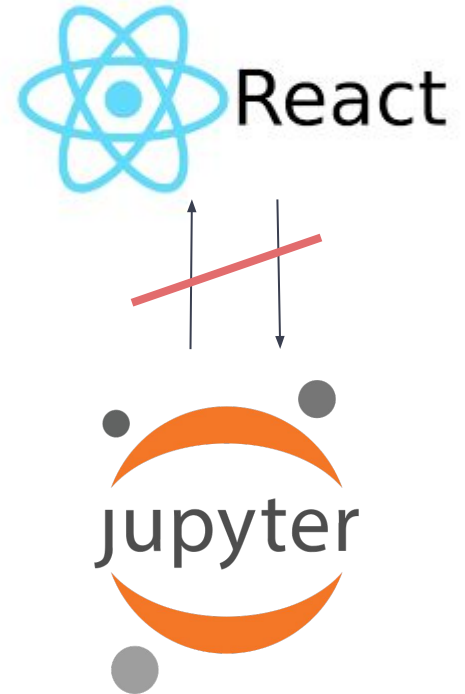
React.js components to build your custom data product with Jupyter

Lightning Talk at FOSDEM 2022
Saturday 5th February 2022

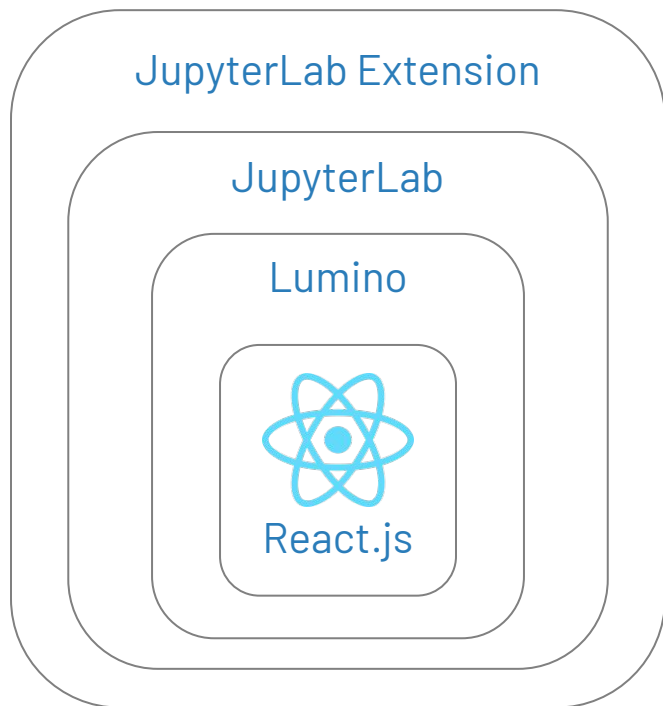


What is the issue?

- ▶ 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

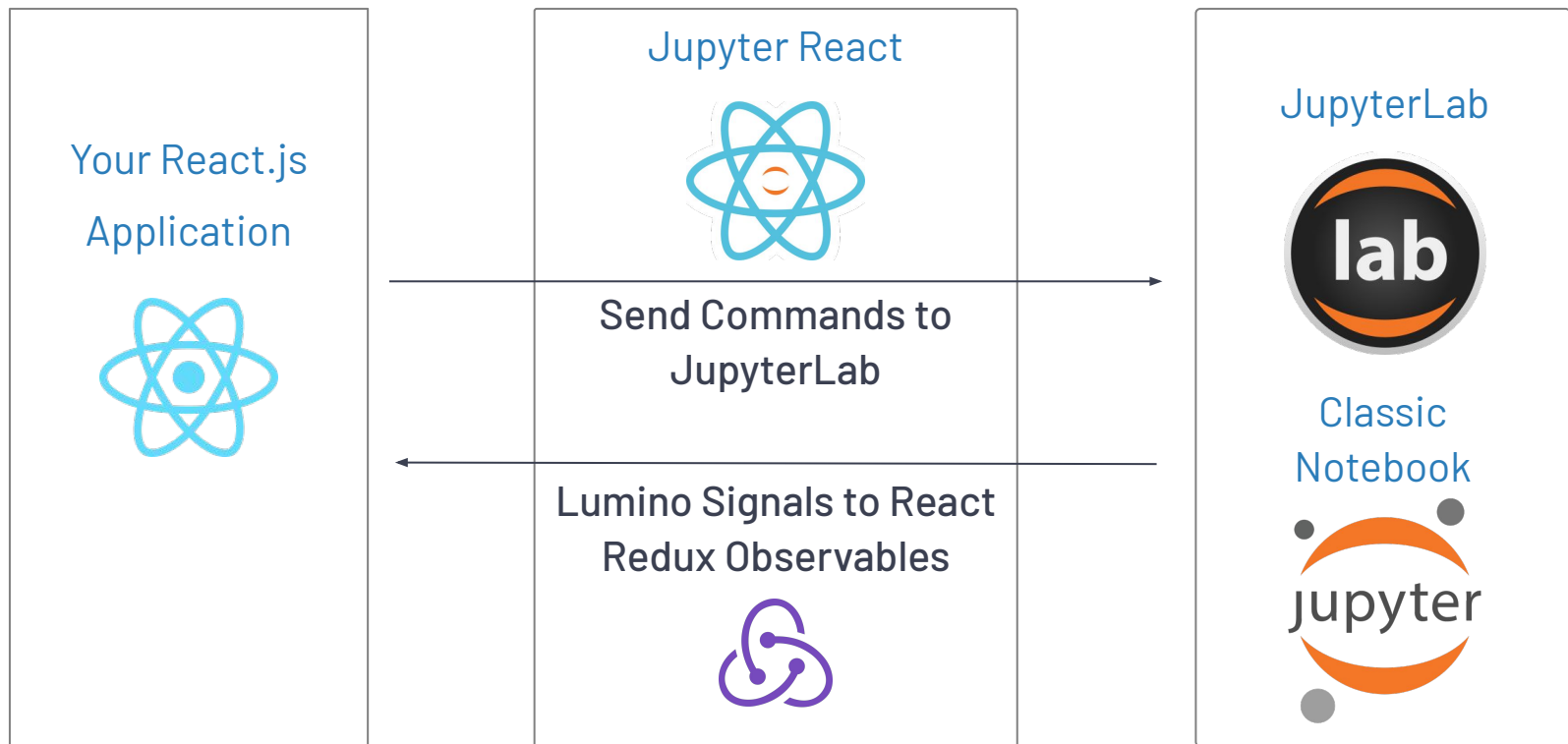
! ! It is a Lumino App and can
not be used by React.js
developers

Why is this an issue?

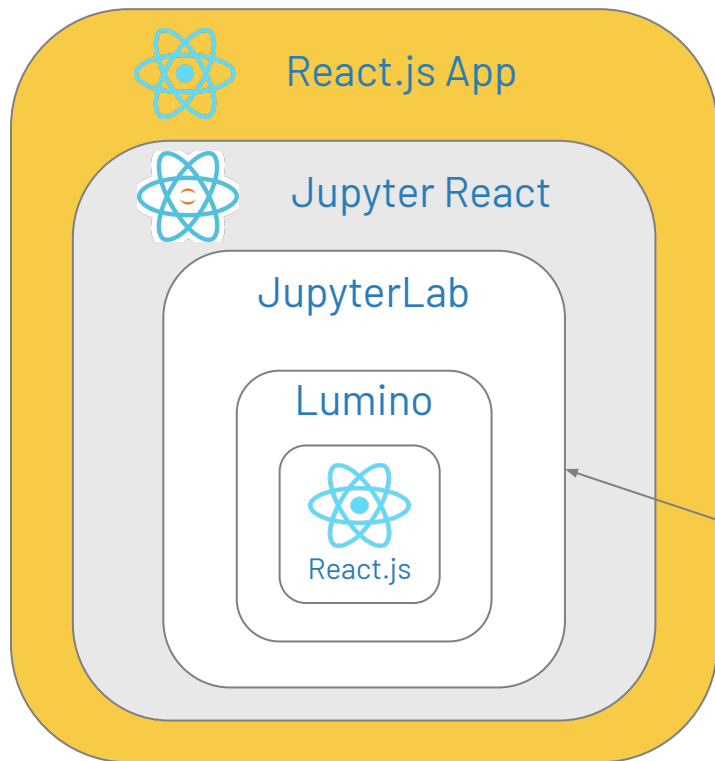
- ▶ You have to fallback to **iframe solutions** to show a composed UI **mixing your existing React UI with JupyterLab**
 - ▷ Those are **separated applications** which does not talk together
- ▶ You **need to learn yet-another UI toolkit (Lumino) to customize JupyterLab**
 - ▷ No “declarative” language
 - ▷ 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



Jupyter React architecture




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

Lumino Command and Signals
 <->
 React Redux Observables

Usage



```

1 <Jupyter terminals={true} collaborative={true}>
2   <Notebook path="test.ipynb"/>
3 </Jupyter>

```



```

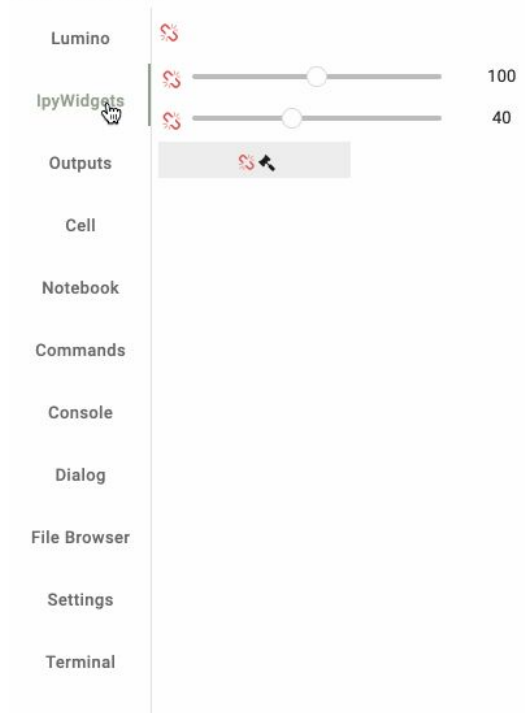
1 export type INotebookProps = {
2   path: string;
3   ipywidgets?: "classic" | "modern";
4   sidebarComponent: (props: any) => JSX.Element;
5   sidebarMargin: number;
6 };
7

```

Gallery

Jupyter React Components Example

© Datalayer, 2021



The screenshot shows a sidebar with the following items and their associated settings:

- Lumino**: A red 'X' icon.
- IpyWidgets**: A red 'X' icon, a slider set to 100, and the number 100.
- Outputs**: A red 'X' icon, a mouse cursor, and a grey background.
- Cell**: No settings.
- Notebook**: No settings.
- Commands**: No settings.
- Console**: No settings.
- Dialog**: No settings.
- File Browser**: No settings.
- Settings**: No settings.
- Terminal**: No settings.

DocuSaurus Example

My Site Tutorial Blog

Tutorial Intro

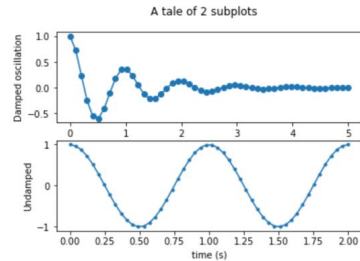
- Tutorial - Basics >
- Tutorial - Extras >

Tutorial Intro

Let's discover DocuSaurus in less than 5 minutes.

Getting Started
Generate a new site
Start your site

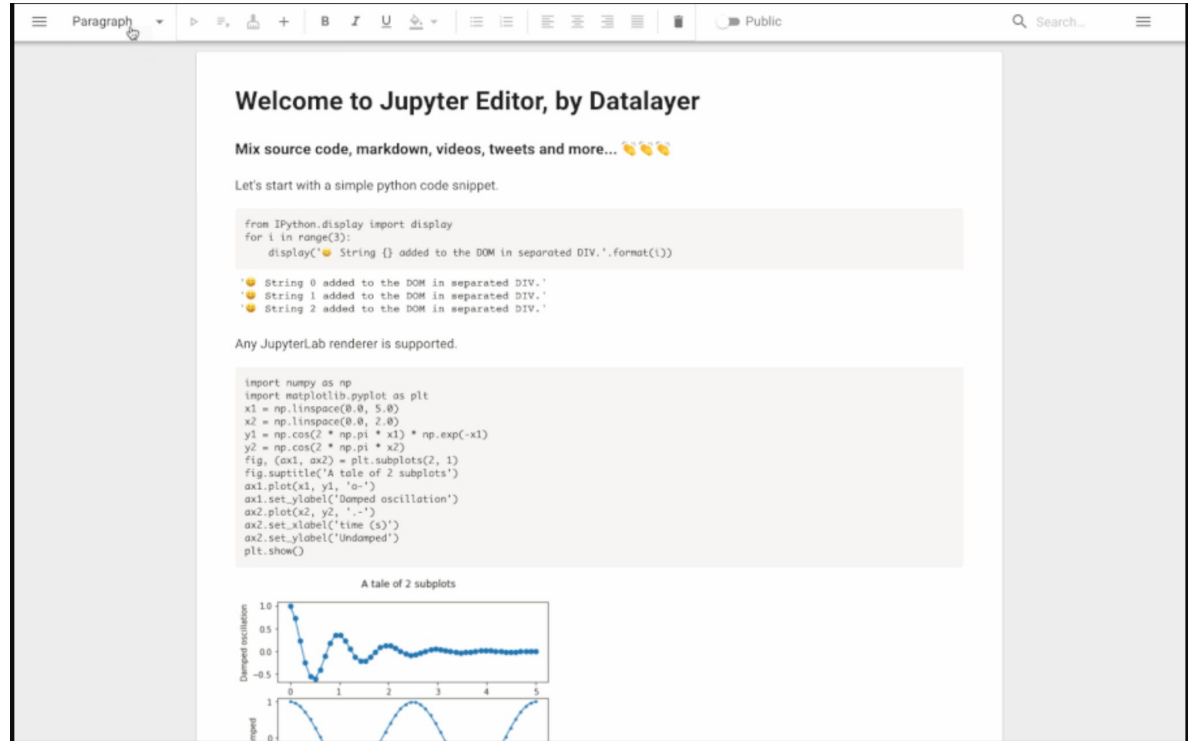
```
[1]: import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
fig, (ax1, ax2) = plt.subplots(2, 1)
fig.suptitle('A tale of 2 subplots')
ax1.plot(x1, y1, 'o-')
ax1.set_ylabel('Damped oscillation')
ax2.plot(x2, y2, '-.')
ax2.set_xlabel('time (s)')
ax2.set_ylabel('Undamped')
plt.show()
```



Getting Started

Slate Example

*Build a
Google-Docs-like
Notebook*



The screenshot shows the Datalayer Jupyter Editor interface. At the top, there is a toolbar with various editing tools and a 'Public' button. The main content area displays a notebook with the following text:

Welcome to Jupyter Editor, by Datalayer

Mix source code, markdown, videos, tweets and more... 🐦🐦🐦

Let's start with a simple python code snippet.

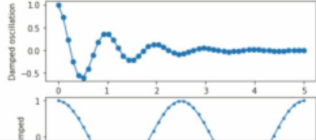
```
from IPython.display import display
for i in range(3):
    display('🍌 String {} added to the DOM in separated DIV.'.format(i))
```

🍌 String 0 added to the DOM in separated DIV.
 🍌 String 1 added to the DOM in separated DIV.
 🍌 String 2 added to the DOM in separated DIV.

Any JupyterLab renderer is supported.

```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
fig, (ax1, ax2) = plt.subplots(2, 1)
fig.suptitle('A tale of 2 subplots')
ax1.plot(x1, y1, 'o-')
ax1.set_ylabel('Damped oscillation')
ax2.plot(x2, y2, '-.')
ax2.set_xlabel('time (s)')
ax2.set_ylabel('Undamped')
plt.show()
```

A tale of 2 subplots



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>)

▶ **Connect** with Jupyter community, developers and users interested in React.js



THANK YOU!

Any questions?

- ▶ eric@datalayer.io
- ▶ @echarles
- ▶ <https://twitter.com/echarles>

- ▶ <https://github.com/datalayer/jupyter-react>
- ▶ <https://github.com/datalayer/jupyter-examples>
- ▶ <https://github.com/datalayer/jupyter-docusaurus>

