
jupyterlab*emailDocumentation*

Release 0.2.3

Tim Paine

Aug 19, 2020

Contents

1	Options	3
2	Install	5
3	Adding templates	7
4	Create email from notebook:	9
5	Attach dataframe as csv or spreadsheet	11
6	Inline LaTeX	13
7	API Documentation	15
	Python Module Index	17
	Index	19

A jupyterlab extension to email notebooks from the browser.

[Build Status](#) [GitHub issues](#) [Coverage](#) [PyPI](#) [PyPI](#) [npm](#)

- Inline notebook as email, with code
- Inline notebook as email, without code
- Send notebook as HTML attachment, with code
- Send notebook as HTML attachment, without code
- Send notebook as PDF attachment, with code
- Send notebook as PDF attachment, without code
- Attach output data as CSV, TSV, PDF, PNG, or Excel Spreadsheet

CHAPTER 2

Install

```
pip install jupyterlab_email
jupyter labextension install jupyterlab_email
jupyter serverextension enable --py jupyterlab_email
```


CHAPTER 3

Adding templates

install the server extension, and add the following to `jupyter_notebook_config.py`

```
c.JupyterLabEmail.smtp_servers = [{'name': 'gmail',  
                                  'domain': 'gmail.com',  
                                  'username': '<YOUR USERNAME>',  
                                  'smtp': 'smtp.gmail.com',  
                                  'port': 465}]
```

Create email from notebook:

Use the function in `jupyterlab_email._email`

```
def make_email(path, model, from_, type='email', template='', code=False, subject='',
               also_attach='none', also_attach_pdf_template='', also_attach_html_
↳template=''):
    """
        path          : path to notebook
        model          : notebook itself (in case deployment strips outputs or
                       notebook not available except through ContentsManager)
        from_         : address to send the email from
        type          : type to convert notebook to
        template      : template to use when converting notebook
        code          : include input cells in notebook
        subject       : subject of email
        also_attach   : also attach pdf/html/both
    """
```

Attach dataframe as csv or spreadsheet

In `jupyterlab_email.attachments`

```
def attach(data, filename, type):
```

Modify `jupyterlab_email.attachments.EXCEL_ENGINE` to use a different excel writer (defaults to `xlsxwriter`)

CHAPTER 6

Inline LaTeX

In `jupyterlab_email.attachments`

```
def latex(expression):  
    import matplotlib.pyplot as plt  
    fig, ax = plt.subplots(figsize=(10, 1))  
    ax.xaxis.set_visible(False)  
    ax.yaxis.set_visible(False)  
    ax.axis('off')  
    plt.text(0, 0.6, r'$$s$$' % expression, fontsize=25)  
    plt.show()
```


`jupyterlab_email.attachments.attach` (*data, filename, type*)

`jupyterlab_email.attachments.latex` (*expression*)

`jupyterlab_email.extension.load_jupyter_server_extension` (*nb_server_app*)

Called when the extension is loaded.

Parameters `nb_server_app` (*NotebookWebApplication*) – handle to the Notebook web-server instance.

```
class jupyterlab_email.handlers.EmailHandler (application:          tor-
                                             nado.web.Application, request: tor-
                                             nado.httputil.HTTPServerRequest,
                                             **kwargs)
```

Bases: `notebook.base.handlers.IPythonHandler`

```
initialize (emails=None, templates=None, user_templates=None, headers=None, footers=None,
            signatures=None, postprocessors=None)
```

```
post ()
```

```
class jupyterlab_email.handlers.EmailsListHandler (application:          tor-
                                                    nado.web.Application, request: tor-
                                                    nado.httputil.HTTPServerRequest,
                                                    **kwargs)
```

Bases: `notebook.base.handlers.IPythonHandler`

```
get ()
```

```
initialize (emails=None, templates=None, user_templates=None, headers=None, footers=None,
            signatures=None, postprocessors=None)
```

`jupyterlab_email.handlers.get_template` (*type, code, template, handler, user_template=""*)

`jupyterlab_email.nbconvert.run` (*to='html', name="", in_="", template="", execute=False, execute_timeout=600*)

```
jupyterlab_email.postprocessors.pivot_pandas_to_excel (soup,  
                                                    show_intermediate_breakdown=False,  
                                                    show_total_breakdown=False)
```

pandas style pivot to excel style pivot formatting for outlook/html

This function is meant to be provided to the email functionality as a postprocessor. It expects a jupyter or pandas exported html table of a dataframe with the following index:

```
example: # a single pivot pt1 = pd.pivot_table(data,  
        value=['col1', 'col2', 'col3'], index=['index'], columns=['col4'], aggfunc='sum', mar-  
        gins=True).stack('col4')
```

```
# here we reindex the table to have the appropriate row ordering pt1 = pt1.reindex(  
    pd.MultiIndex( levels=[['COL1', 'COL2', 'All'],  
        ['ROW1', 'ROW2', 'ALL']],  
        labels=[[0, 0, 0, 1, 1, 1, 2], # This is the key, changing the label order (2-ALL) [2, 0, 1, 2, 0, 1,  
        2],  
        names=['', ''], sortorder=0) ).fillna(0)
```

show_intermediate_breakdown -> intermediate sumations to be shown? show_total_breakdown -> total sumations to be shown?

j

jupyterlab_email, 15
jupyterlab_email.attachments, 15
jupyterlab_email.extension, 15
jupyterlab_email.handlers, 15
jupyterlab_email.nbconvert, 15
jupyterlab_email.postprocessors, 15

A

`attach()` (in module `jupyterlab_email.attachments`), 15

E

`EmailHandler` (class in `jupyterlab_email.handlers`), 15

`EmailsListHandler` (class in `jupyterlab_email.handlers`), 15

G

`get()` (`jupyterlab_email.handlers.EmailsListHandler` method), 15

`get_template()` (in module `jupyterlab_email.handlers`), 15

I

`initialize()` (`jupyterlab_email.handlers.EmailHandler` method), 15

`initialize()` (`jupyterlab_email.handlers.EmailsListHandler` method), 15

J

`jupyterlab_email` (module), 15

`jupyterlab_email.attachments` (module), 15

`jupyterlab_email.extension` (module), 15

`jupyterlab_email.handlers` (module), 15

`jupyterlab_email.nbconvert` (module), 15

`jupyterlab_email.postprocessors` (module), 15

L

`latex()` (in module `jupyterlab_email.attachments`), 15

`load_jupyter_server_extension()` (in module `jupyterlab_email.extension`), 15

P

`pivot_pandas_to_excel()` (in module `jupyterlab_email.postprocessors`), 15

`post()` (`jupyterlab_email.handlers.EmailHandler` method), 15

R

`run()` (in module `jupyterlab_email.nbconvert`), 15