
Jupyterlab extension workshop

— #jupytercon2020 —

What are Server Extensions

- Written in Python
- Module that extends the Server's REST API/endpoints
- Can provide a backend for an extension or server side processing
- Documentation
 - <https://jupyter-server.readthedocs.io/en/latest/developers/extensions.html>
- Examples
 - https://github.com/jupyter/jupyter_server/tree/master/examples/simple
 - <https://github.com/jupyterlab/jupyterlab-latex>

Starting files

- **setup.py**
 - Tells pip how to install a package, includes info like the url your API can be accessed by
- Directory mybutton
 - **__init__.py**
 - Sets up package - jupyterlab will look for this to find metadata about the server extension
 - **handlers.py**
 - Contains the code that handles different requests

Understanding the “Hello World” extension handler

setup.py

```
from setuptools import setup
setup(
    name="mybutton",
    include_package_data=True,
    data_files=[
        (
            "etc/jupyter/jupyter_server_config.d",
            ["jupyter-config/jupyter_server_config.d/mybutton.json"]
        ),
    ]
)
```

Understanding the “Hello World” extension handler

mybutton/_init_.py

```
from .handlers import TutorialHandler

def _jupyter_server_extension_points():
    return [{
        "module": "mybutton"
    }]

def load_jupyter_server_extension(server_app):
    handlers = [("/mybutton/hello", TutorialHandler)]
    server_app.web_app.add_handlers(".*$", handlers)
```

Understanding the “Hello World” extension handler

mybutton/handlers.py

```
from jupyter_server.base.handlers import JupyterHandler
import tornado

class TutorialHandler(JupyterHandler):
    @tornado.web.authenticated
    def get(self):
        self.write('HELLO WORLD')
```

Testing the “Hello World” handler

```
pip install jupyterlab==2.2.8 jupyter_server
```

```
pip install -e . → calls setup.py and installs the python source code
```

```
jupyter serverextension enable mybutton → enables the package to become a server extension
```

debugging tip if the enable fails, try running

```
jupyter lab --ServerApp.jpserver_extensions={"mybutton": True} --debug
```

```
jupyter lab → starts jupyterlab and launch it in your browser
```

Go to URL: <http://localhost:8888/mybutton/hello>

Exercise - Calling the “Hello World” handler from mybutton

Goal: Change the onClick() in src/button.ts to make an alert that shows the text from a GET request to your handler URL

Useful URLs / examples:

- Documentation for ServerConnection
<https://jupyterlab.github.io/jupyterlab/modules/services/src/index.serverconnection.html>
- Documentation for Response
https://microsoft.github.io/PowerBI-JavaScript/interfaces/node_modules/typedoc/node_modules/typescript/lib/lib.dom.d.response.html#text
- Example of ServerConnection being used
<https://github.com/jupyterlab/jupyterlab/blob/7204b461515890d86cba5a3b51832708ae265e8a/packages/services/src/kernelspec/restapi.ts>

*Development tips:

- If you see a class being used in examples, look it up in the documentation and understand the API before using it
- Try searching for all times ServerConnection is used in the JupyterLab repository