CTBA

Pamela Schlosser

2025-08-11

Table of contents

1 How to Deploy a Dash App to Render.com (and Publish HTML via docs/)

2 Overview

- This guide explains how to:
  - Prepare and run a Dash app locally
  - Deploy it to Render.com using Gunicorn
  - Fix common deployment issues
  - Publish an HTML version of a Python file (or this guide) via GitHub Pages
  - Use a docs/ folder to host HTML with GitHub Pages

3 Live Example (Embedded Iframe)

- This shows a hosted Dash app inside the HTML output of this page.
- Replace the src URL with your own Render app link if different.

```
<iframe src="https://ctba-oror.onrender.com/" title="Live Dash App on Render" style="width:1
```

1) Prepare Your Dash App

- Ensure your main Python file defines: server = app.server
- The Gunicorn start command on Render must match your filename and server variable exactly

```
from dash import Dash, html

app = Dash(__name__)
server = app.server  # Required for Gunicorn in production

wm_green = "#115740"

app.layout = html.Div([
    html.H1("Hello from Dash on Render!", style={"color": wm_green, "textAlign": "center"})
])

if __name__ == "__main__":
    app.run(debug=True)
```

```
<IPython.lib.display.IFrame at 0x1a132927560>
```

- Example mapping:
  - File: LiveDash.py –> Start command: gunicorn LiveDash:server
  - File: electricity.py –> Start command: gunicorn electricity:server

2) Add Required Files

- Create a requirements.txt listing dependencies (add any others you import).

```
dash
gunicorn
plotly
```

3) Run Locally

- Create and activate a virtual environment
- Install dependencies
- Test both dev mode and production mode

```
python -m venv .venv
.venv\Scripts\activate    # Windows

pip install -r requirements.txt
python your_filename.py  # dev mode
gunicorn your_filename:server  # production test
```

4) Push to GitHub

```
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/USERNAME/REPO.git
git push -u origin main
```

5) Deploy on Render.com

- Create a New Web Service on Render and connect your GitHub repo
- In Settings, set the Build command to: pip install -r requirements.txt
- In Settings, set the Start command to: gunicorn your_filename:server
- Click Deploy and open the provided URL

6) Troubleshooting

- ModuleNotFoundError: Check filename and server variable (your_filename:server)
- Changes not showing: Clear build cache and redeploy
- Missing packages: Confirm all imports are in requirements.txt
- Port errors: Don't hardcode PORT—Render sets it automatically

7) Publish HTML via docs/ on GitHub Pages

- Render this Quarto file to docs/ for GitHub Pages:

```
quarto render dash_to_render.qmd --to html --output-dir docs
git add docs
git commit -m "Publish HTML"
git push
```

GitHub Pages setup:

- Settings → Pages
- Source: Deploy from a branch
- Branch: main
- Folder: /docs
- Save

4 Your site will be available at:

- https://YOUR-USERNAME.github.io/YOUR-REPO/

4.1 Appendix A: Render a Python file to HTML with Quarto

Quarto can render a Python script (with outputs) to HTML.

```
quarto render your_script.py --to html --output-dir docs
```

5 Electricity Prices Dashboard

5.1 Introduction

This dashboard shows **U.S. state-level electricity prices** using a choropleth map.
You can interact with the year slider to see changes over time.

Note

To view the live dashboard inside Quarto, you'll need to run this as an interactive document
(`quarto preview`) rather than a static HTML render.

5.2 Run the Dash App

5.3 Live Dash App (embedded)

This Quarto document **launches the Dash server in the background** when you run `quarto preview`, then embeds it below.

Tip

Run this document with:
"'bash quarto preview intro.qmd

```python
# Run the app from the electricity.py file
import runpy
runpy.run_path("electricity.py")
```

```
{'__name__': '<run_path>',
 '__doc__': None,
 '__package__': '',
 '__loader__': None,
 '__spec__': None,
 '__file__': 'electricity.py',
 '__cached__': None,
 '__builtins__': {'__name__': 'builtins',
  '__doc__': "Built-in functions, types, exceptions, and other objects.\n\nThis module provic
  '__package__': '',
  '__loader__': _frozen_importlib.BuiltinImporter,
  '__spec__': ModuleSpec(name='builtins', loader=<class '_frozen_importlib.BuiltinImporter'>
  '__build_class__': <function __build_class__>,
  '__import__': <function __import__(name, globals=None, locals=None, fromlist=(), level=0)>
  'abs': <function abs(x, /)>,
  'all': <function all(iterable, /)>,
  'any': <function any(iterable, /)>,
  'ascii': <function ascii(obj, /)>,
  'bin': <function bin(number, /)>,
  'breakpoint': <function breakpoint>,
  'callable': <function callable(obj, /)>,
  'chr': <function chr(i, /)>,
  'compile': <function compile(source, filename, mode, flags=0, dont_inherit=False, optimize=
  'delattr': <function delattr(obj, name, /)>,
  'dir': <function dir>,
  'divmod': <function divmod(x, y, /)>,
  'eval': <function eval(source, globals=None, locals=None, /)>,
  'exec': <function exec(source, globals=None, locals=None, /, *, closure=None)>,
  'format': <function format(value, format_spec='', /)>,
  'getattr': <function getattr>,
  'globals': <function globals()>,
  'hasattr': <function hasattr(obj, name, /)>,
  'hash': <function hash(obj, /)>,
  'hex': <function hex(number, /)>,
  'id': <function id(obj, /)>,
  'input': <bound method Kernel.raw_input of <ipykernel.ipkernel.IPythonKernel object at 0x00
  'isinstance': <function isinstance(obj, class_or_tuple, /)>,
  'issubclass': <function issubclass(cls, class_or_tuple, /)>,
  'iter': <function iter>,
  'aiter': <function aiter(async_iterable, /)>,
  'len': <function len(obj, /)>,
  'locals': <function locals()>,
  'max': <function max>,
  'min': <function min>,
  'next': <function next>,
  'anext': <function anext>,
  'oct': <function oct(number, /)>,
 'ord': <function ord(c, /)>,
 'pow': <function pow(base, exp, mod=None)>,
 'print': <function print(*args, sep=' ', end='\n', file=None, flush=False)>,
 'repr': <function repr(obj, /)>,
 'round': <function round(number, ndigits=None)>,
 'setattr': <function setattr(obj, name, value, /)>,
 'sorted': <function sorted(iterable, /, *, key=None, reverse=False)>,
 'sum': <function sum(iterable, /, start=0)>,
 'vars': <function vars>,
 'None': None,
 'Ellipsis': Ellipsis,
 'NotImplemented': NotImplemented,
 'False': False,
 'True': True,
 'bool': bool,
 'memoryview': memoryview,
 'bytearray': bytearray,
 'bytes': bytes,
 'classmethod': classmethod,
 'complex': complex,
 'dict': dict,
 'enumerate': enumerate,
 'filter': filter,
 'float': float,
 'frozenset': frozenset,
 'property': property,
 'int': int,
 'list': list,
 'map': map,
 'object': object,
 'range': range,
 'reversed': reversed,
 'set': set,
 'slice': slice,
 'staticmethod': staticmethod,
 'str': str,
 'super': super,
 'tuple': tuple,
 'type': type,
 'zip': zip,
 '__debug__': True,
 'BaseException': BaseException,
 'BaseExceptionGroup': BaseExceptionGroup,
 'Exception': Exception,
 'GeneratorExit': GeneratorExit,
 'KeyboardInterrupt': KeyboardInterrupt,
 'SystemExit': SystemExit,
 'ArithmeticError': ArithmeticError,
 'AssertionError': AssertionError,
 'AttributeError': AttributeError,
 'BufferError': BufferError,
 'EOFError': EOFError,
 'ImportError': ImportError,
 'LookupError': LookupError,
 'MemoryError': MemoryError,
 'NameError': NameError,
 'OSError': OSError,
 'ReferenceError': ReferenceError,
 'RuntimeError': RuntimeError,
 'StopAsyncIteration': StopAsyncIteration,
 'StopIteration': StopIteration,
 'SyntaxError': SyntaxError,
 'SystemError': SystemError,
 'TypeError': TypeError,
 'ValueError': ValueError,
 'Warning': Warning,
 'FloatingPointError': FloatingPointError,
 'OverflowError': OverflowError,
 'ZeroDivisionError': ZeroDivisionError,
 'BytesWarning': BytesWarning,
 'DeprecationWarning': DeprecationWarning,
 'EncodingWarning': EncodingWarning,
 'FutureWarning': FutureWarning,
 'ImportWarning': ImportWarning,
 'PendingDeprecationWarning': PendingDeprecationWarning,
 'ResourceWarning': ResourceWarning,
 'RuntimeWarning': RuntimeWarning,
 'SyntaxWarning': SyntaxWarning,
 'UnicodeWarning': UnicodeWarning,
 'UserWarning': UserWarning,
 'BlockingIOError': BlockingIOError,
 'ChildProcessError': ChildProcessError,
 'ConnectionError': ConnectionError,
 'FileExistsError': FileExistsError,
 'FileNotFoundError': FileNotFoundError,
 'InterruptedError': InterruptedError,
 'IsADirectoryError': IsADirectoryError,
 'NotADirectoryError': NotADirectoryError,
 'PermissionError': PermissionError,
 'ProcessLookupError': ProcessLookupError,
 'TimeoutError': TimeoutError,
 'IndentationError': IndentationError,
 'IndexError': IndexError,
 'KeyError': KeyError,
 'ModuleNotFoundError': ModuleNotFoundError,
 'NotImplementedError': NotImplementedError,
 'RecursionError': RecursionError,
 'UnboundLocalError': UnboundLocalError,
 'UnicodeError': UnicodeError,
 'BrokenPipeError': BrokenPipeError,
 'ConnectionAbortedError': ConnectionAbortedError,
 'ConnectionRefusedError': ConnectionRefusedError,
 'ConnectionResetError': ConnectionResetError,
 'TabError': TabError,
 'UnicodeDecodeError': UnicodeDecodeError,
 'UnicodeEncodeError': UnicodeEncodeError,
 'UnicodeTranslateError': UnicodeTranslateError,
 'ExceptionGroup': ExceptionGroup,
 'EnvironmentError': OSError,
 'IOError': OSError,
 'WindowsError': OSError,
 'open': <function _io.open(file, mode='r', buffering=-1, encoding=None, errors=None, newli
 'copyright': Copyright (c) 2001-2023 Python Software Foundation.
 All Rights Reserved.

 Copyright (c) 2000 BeOpen.com.
 All Rights Reserved.

 Copyright (c) 1995-2001 Corporation for National Research Initiatives.
 All Rights Reserved.

 Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
 All Rights Reserved.,
 'credits':     Thanks to CWI, CNRI, BeOpen.com, Zope Corporation and a cast of thousands
     for supporting Python development.  See www.python.org for more information.,
 'license': MIT License

 Copyright (c) 2025 SchlosserPG
```

```
 Permission is hereby granted, free of charge, to any person obtaining a copy
 of this software and associated documentation files (the "Software"), to deal
 in the Software without restriction, including without limitation the rights
 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 copies of the Software, and to permit persons to whom the Software is
 furnished to do so, subject to the following conditions:

 The above copyright notice and this permission notice shall be included in all
 copies or substantial portions of the Software.

 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 SOFTWARE.,
 'help': Type help() for interactive help, or help(object) for help about object.,
 'execfile': <function _pydev_bundle._pydev_execfile.execfile(file, glob=None, loc=None)>,
 'runfile': <function _pydev_bundle.pydev_umd.runfile(filename, args=None, wdir=None, names
 '__IPYTHON__': True,
 'display': <function IPython.core.display_functions.display(*objs, include=None, exclude=N
 'get_ipython': <bound method InteractiveShell.get_ipython of <ipykernel.zmqshell.ZMQIntera
'dash': <module 'dash' from 'C:\\Users\\pamel\\Documents\\BUAD-CTBA\\Code\\.venv\\Lib\\site
'dcc': <module 'dash.dcc' from 'C:\\Users\\pamel\\Documents\\BUAD-CTBA\\Code\\.venv\\Lib\\s
'html': <module 'dash.html' from 'C:\\Users\\pamel\\Documents\\BUAD-CTBA\\Code\\.venv\\Lib\
'Input': dash.dependencies.Input,
'Output': dash.dependencies.Output,
'pd': <module 'pandas' from 'C:\\Users\\pamel\\Documents\\BUAD-CTBA\\Code\\.venv\\Lib\\site
'px': <module 'plotly.express' from 'C:\\Users\\pamel\\Documents\\BUAD-CTBA\\Code\\.venv\\L
'df':        year  month state      sectorName  customers  price    revenue  \
0      2001      1    WY     all sectors        NaN   4.31   48.12840
1      2001      1    WY      commercial        NaN   5.13   12.67978
2      2001      1    WY      industrial        NaN   3.26   19.60858
3      2001      1    WY           other        NaN   4.75    0.76868
4      2001      1    WY     residential        NaN   6.01   15.07136
...     ...    ...   ...             ...        ...    ...        ...
81710  2024      1    AR     all sectors  1717720.0   9.63  442.98773
81711  2024      1    AR      commercial   208669.0  10.26   97.79467
81712  2024      1    AR      industrial    34951.0   7.08  109.92656
81713  2024      1    AR     residential  1474098.0  11.24  235.26399
81714  2024      1    AR  transportation        2.0  12.70    0.00252
```

```
            sales
0      1116.17208
1       247.08691
2       602.30484
3        16.17442
4       250.60591
...           ...
81710  4598.63147
81711   953.02154
81712  1553.02838
81713  2092.56172
81714     0.01984

[81715 rows x 8 columns],
'app': <dash.dash.Dash at 0x2b03fe466c0>,
'update_map': <function <run_path>.update_map(selected_year)>}
```

5.3.1 How it works

- The `runpy.run_path("electricity.py")` call will execute the file directly in the Quarto execution environment.
- If you want to **serve the app locally** and not block the document rendering, you could instead run the script in the terminal: