



Get started creating your own Dash components here.

☆ 281 stars

🍴 193 forks

👁 44 watching

🔑 Branches

🏠 Activity

📁 Custom properties

🏷 Tags

🌐 Public repository

T4rk1n

Merge pull request #164 from neetishsingh/master

788d21d · 9 months ago


📁 .circleci	update test names	9 months ago
📁 .github	Replace all instances of plot.ly with pl...	5 years ago
📁 hooks	Revert back use_async usage and add...	2 years ago
📁 tests	fix test	3 years ago
📁 {{cookiecutter.project_shortna...	Merge branch 'master' into master	9 months ago
📄 .gitignore	Git ignore cache files	4 years ago
📄 CHANGELOG.md	🐛 first pass	7 years ago
📄 README.md	Update README.md with "maintaine...	last year
📄 cookiecutter.json	Revert back use_async usage and add...	2 years ago
📄 pytest.ini	Fix case of words and standardize the...	6 years ago

📖 README

# Dash Component Boilerplate

This repository contains a [Cookiecutter](#) template to create your own Dash components.

- New to Dash? Check out the [official Dash Documentations](#)
- If it's the first time creating a Dash component, start with our [React guide for Python developers](#)
- Need help with your component? Create a post on the [Dash Community Forum](#)

Maintained by  plotly

# Usage

To use this boilerplate:

1. Install the requirements:

```
$ pip install cookiecutter
$ pip install virtualenv
```



[Node.js/npm is also required.](#)

2. Run cookiecutter on the boilerplate repo:

```
$ cookiecutter gh:plotly/dash-component-boilerplate
```



3. Answer the questions about the project.

- `project_name` : This is the "human-readable" name of your project. For example, "Dash Core Components".
- `project_shortname` : is derived from the project name, it is the name of the "Python library" for your project. By default, this is generated from your `project_name` by lowercasing the name and replacing spaces & - with underscores. For example, for "Dash Core Components" this would be "dash\_core\_components".
- `component_name` : This is the name of the initial component that is generated. As a JavaScript class name it should be in PascalCase. defaults to the PascalCase version of `project_shortname` .
- `jl_prefix` : Optional prefix for Julia components. For example, `dash_core_components` uses "dcc" so the Python `dcc.Input` becomes `dccInput` in Julia, and `dash_table` uses "dash" to make `dashDataTable` .
- `r_prefix` : Optional prefix for R components. For example, `dash_core_components` uses "dcc" so the Python `dcc.Input` becomes `dccInput` in R, and `dash_table` uses "dash" to make `dashDataTable` .
- `author_name` and `author_email` : for package.json and DESCRIPTION (for R) metadata.
- `github_org` : If you plan to push this to GitHub, enter the organization or username that will own it (for URLs to the project homepage and bug report page)
- `description` : the project description, included in package.json.
- `license` : License type for the component library. Plotly recommends the MIT license, but you should read the generated LICENSE file carefully to make sure this is right for you.
- `publish_on_npm` : Set to false to only serve locally from the package data.
- `install_dependencies` : Set to false to only generate the project structure.

4. The project will be generated in a folder named with your `project_shortname` .

5. Follow the directions in the generated README to start developing your new Dash component.

Installing the dependencies can take a long time. They will be installed in a folder named `venv` , created by `virtualenv`. This ensures that dash is installed to generate the components in the `build:backends` script of the generated `package.json` .

## Advanced customization

### Shared cache groups for async chunks

Shared async chunks for code that repeats across multiple async chunks is already supported through our custom `webpack.config.js` optimizations. You can leverage it by manually the path of

```
{{cookiecutter.project_shortname}}-shared.js to _js_dist inside
{{cookiecutter.project_shortname}}/_init__.py (as well as the associated external URL).
```

## More Resources

- Found a bug or have a feature request? [Create an issue](#)
- Watch the [component boilerplate repository](#) to stay informed of changes to our components.
- To get a feel for what's involved in creating a component, read the [README.md generated by this cookiecutter](#)
- Want something more visual? Check out [this asciinema](#) of how to create a dash component from this boilerplate.
- Examples of Dash component libraries include:
  - [dash-core-components](#)
  - [dash-html-components](#)

## Releases

No releases published

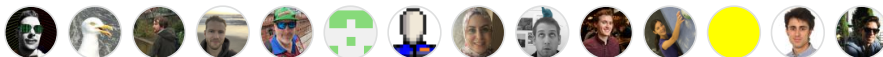
## Sponsor this project

<https://plotly.com/products/consulting-and-oem/>

## Packages

No packages published

## Contributors 29



[+ 15 contributors](#)

## Languages

● Python 58.0%   ● JavaScript 41.2%   ● HTML 0.8%